L15 - Integrated Motion on Ethernet/IP

For Classroom Use Only!

LISTEN. THINK. SOLVE.®

Allen-Bradley • Rockwell Software

Rockwell Automation

Important User Information

This documentation, whether, illustrative, printed, “online” or electronic (hereinafter “Documentation”) is intended for use only as a learning aid when using Rockwell Automation approved demonstration hardware, software and firmware. The Documentation should only be used as a learning tool by qualified professionals.

The variety of uses for the hardware, software and firmware (hereinafter “Products”) described in this Documentation, mandates that those responsible for the application and use of those Products must satisfy themselves that all necessary steps have been taken to ensure that each application and actual use meets all performance and safety requirements, including any applicable laws, regulations, codes and standards in addition to any applicable technical documents.

In no event will Rockwell Automation, Inc., or any of its affiliate or subsidiary companies (hereinafter “Rockwell Automation”) be responsible or liable for any indirect or consequential damages resulting from the use or application of the Products described in this Documentation. Rockwell Automation does not assume responsibility or liability for damages of any kind based on the alleged use of, or reliance on, this Documentation.

No patent liability is assumed by Rockwell Automation with respect to use of information, circuits, equipment, or software described in the Documentation.

Except as specifically agreed in writing as part of a maintenance or support contract, equipment users are responsible for:

- properly using, calibrating, operating, monitoring and maintaining all Products consistent with all Rockwell Automation or third-party provided instructions, warnings, recommendations and documentation;
- ensuring that only properly trained personnel use, operate and maintain the Products at all times;
- staying informed of all Product updates and alerts and implementing all updates and fixes; and • all other factors affecting the Products that are outside of the direct control of Rockwell Automation.

Reproduction of the contents of the Documentation, in whole or in part, without written permission of Rockwell Automation is prohibited.

Throughout this manual we use the following notes to make you aware of safety considerations:

WARNING

Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.

IMPORTANT

Identifies information that is critical for successful application and understanding of the product.

ATTENTION

Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you:

- identify a hazard
- avoid a hazard
- recognize the consequence

SHOCK HAZARD

Labels may be located on or inside the drive to alert people that dangerous voltage may be present.

BURN HAZARD

Labels may be located on or inside the drive to alert people that surfaces may be dangerous temperatures.

Integrated Motion on Ethernet/IP

Contents

Before you begin 6

About this lab 6

Tools & prerequisites 6

Network Setup 7

About the CompactLogix Demo 9

About the Kinetix 5500 and PF527 3-Axis Demo 10

Demonstration (Estimated Time 15 minutes) 10

Launch Studio 5000 and Open Application Files 11

Open and Run the HMI Application 13

Start and Stop the Machine 14

CIP Motion Axis Faceplate Manual Control 15
Lab 1: Basic Configuration of an Integrated Motion System (Estimated Time 20 Minutes) 17
Open the Application File 17
Hardware and Network Considerations 22
Add Your Drive Hardware 24
Configure Axis Properties 32
Save and Download Your Motion Project 43
Lab 2: Axis Commissioning – Hookup Test and Autotune (Estimated Time 10 Minutes) 46
Axis Hookup Tests 46
Axis Autotune 49
Common Faults Encountered While Tuning 53
Lab 3: Using Motion Direct Commands (Estimated Time 10 Minutes) 55
Jogging an Axis Using Motion Direct Commands 55
Varying the Speed of the Axis Using a Motion Direct Command 57
Stop the Axis Using a Motion Direct Command 58
Lab 4: Adding an HMI (Estimated Time 15 Minutes) 60
Switch to the HMI Application 60
Faceplate Operation 61
Lab 5: Troubleshooting Techniques (Estimated Time 10 Minutes) 69
Diagnostic Capabilities of Logix Designer 69
APPENDIX (**_Optional_**) 74
Appendix A: Logix Coordinated Motion 75
Creating the Coordinate System 75
Add a Motion Coordinated Linear Move (MCLM) and a Motion Coordinated Circular Move (MCCM) 78
Execute the Coordinated Motion Profile 82
Appendix B: Master Driven Speed Control (MDAC) Lab 86
Execute the MAM Instruction in Classic Mode 86
Execute and Verify a MAM in MDSC mode using Units per MasterUnit 92
Execute and Verify a MAM in MDSC mode using MasterUnits 96
Appendix C: PCAM Rotary Knife Application Lab 99
Overview of the Example Machine 100
Review Code and Execute a Cam Profile 103
Designing a Basic Cam Table 109
Appendix D: Tuning Techniques Lab 114
Configuring the Default Tuning Configuration 116
Identify and Compensate for Mechanical Resonances 123
Appendix E: Network (CIP) Safety 124
Configure a Network Safety Drive 128
Write Program Code 134

Appendix F: Multiplexing Introduction 138
Using Multiplexing to Optimize Performance 138
IAB Info 141

Before you begin

Prerequisite is to be familiar with Logix Designer software and programming. When the computer is booted, a FactoryTalk View ME Station program will start. You will use that Human Machine Interface (HMI) throughout the lab. If you close it at any time, you will need to open it again from the C:\Lab Files directory in order to operate the lab correctly.

Logix Designer

Studio 5000 is a modular framework for engineering collaboration with plug-ins for specific engineering tasks. For example, there will be a core plug-in that will be used for developing projects for Logix controllers. This plug-in is referred to as Logix Designer. Logix Designer brings the existing RSLogix 5000 user interface into the Studio 5000 environment which will introduce new shared components. These components will bring even more power, flexibility, and organization to the Logix design environment. Studio 5000 will be required for all Logix controllers that are running version 21 firmware or greater.

About this lab

You will be introduced to Logix Designer software environment as the single software tool used by the Rockwell Automation Integrated Motion Solution for configuration, programming, and troubleshooting, as well as the inherent ease with which you can define your motion process.

This lab exercise demonstrates the following concepts of Integrated Motion on EtherNet/IP:

- Time efficient nature of using an Integrated Motion solution
- Benefits of Integrated Motion on EtherNet/IP
- Power and performance-oriented nature of the Integrated Motion solution
- Ease of motion system setup utilizing the ‘Drives & Motion Accelerator Toolkit’

You will see how easy it is to create an Integrated Motion Solution by doing the following:

- Creating and configuring motion axes using Logix Designer
- Learning basic motion-direct commands
- Utilizing the ‘Drives & Motion Accelerator Toolkit’ to speed programming of your motion application
- Learning some basic troubleshooting techniques
- Being introduced to the advanced diagnostic tools available in the controller
- Learn advanced motion topics such as camming, tuning, and coordinated motion.

During this lab you will be able to understand how Logix Designer can help you reduce the number of hardware and software components as well as the flexibility associated with information/data access in the control system.

Tools & prerequisites

For this hands-on lab, we have provided you with the following materials that will allow you to complete the labs in this workbook.

Software:

- Logix Designer v24.00
- FactoryTalk View ME Station v7.00
- RSLinx Classic v3.61

Hardware:

- Computer with Windows 7 operating system
- CompactLogix 1769-L36ERM Demo (DEMO-CMXL361) ▪ Kinetix 5500 3-axis Demo w/ PowerFlex 527 (09P096G) ▪ Ethernet Patch Cables
- 3 x RJ45 to RJ45 (2m length)
- 2 x RJ45 to RJ45 (1m length)

Required Files:

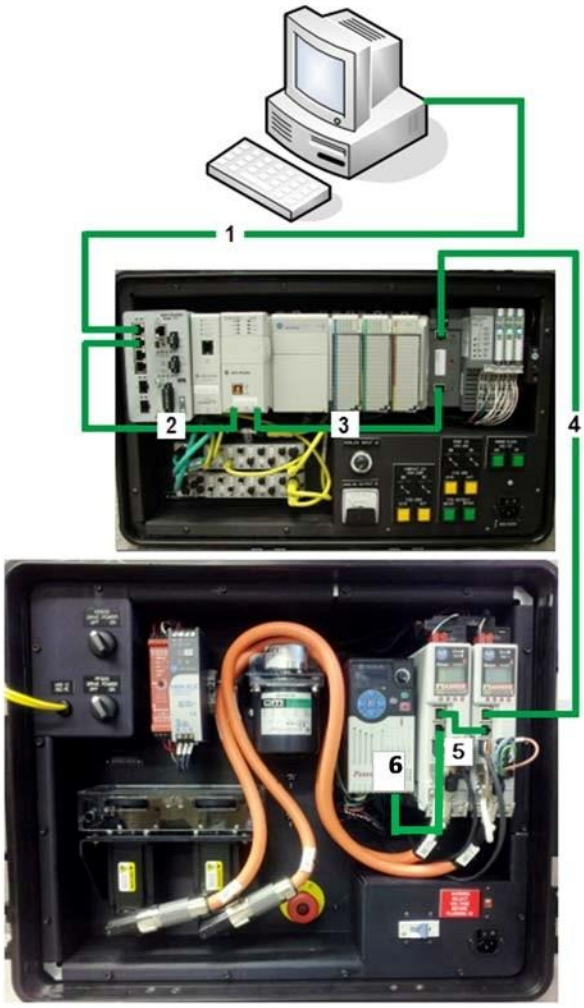
- Integ_Motion_K5500_PF527_Complete.ACD
- Integ_Motion_K5500_PF527_Base.ACD
- Integ_Motion_K5500_PF527_ViewME.MER

Network Setup

Note: This is the recommended configuration for the lab, however due to the variable nature of EtherNet/IP topologies, many other configurations will work.

1	Ethernet Switch, Port 1	to Computer
2	Ethernet Switch, Port2	to Processor, Port 1
3	Processor, Port 2	to POINT I/O, Port 2
4	POINT I/O, Port 1	To Kinetix 5500 Drive 02, Port 2
5	Kinetix 5500 Drive 02, Port 1	To Kinetix 5500 Drive 01, Port 2
6	Kinetix 5500 Drive 01, Port 1	To PF527 Drive 03, Port 2

Computer	192.168.1.1
CompactLogix Processor	192.168.1.12
POINT I/O Ethernet Adapter	192.168.1.8



ArmorBlock Input Module	192.168.1.9 (not used)
ArmorBlock Output Module	192.168.1.10 (not used)
Kinetix 5500 Drive01	192.168.1.24
Kinetix 5500 Drive02	192.168.1.25
PowerFlex 527 Drive03	192.168.1.26

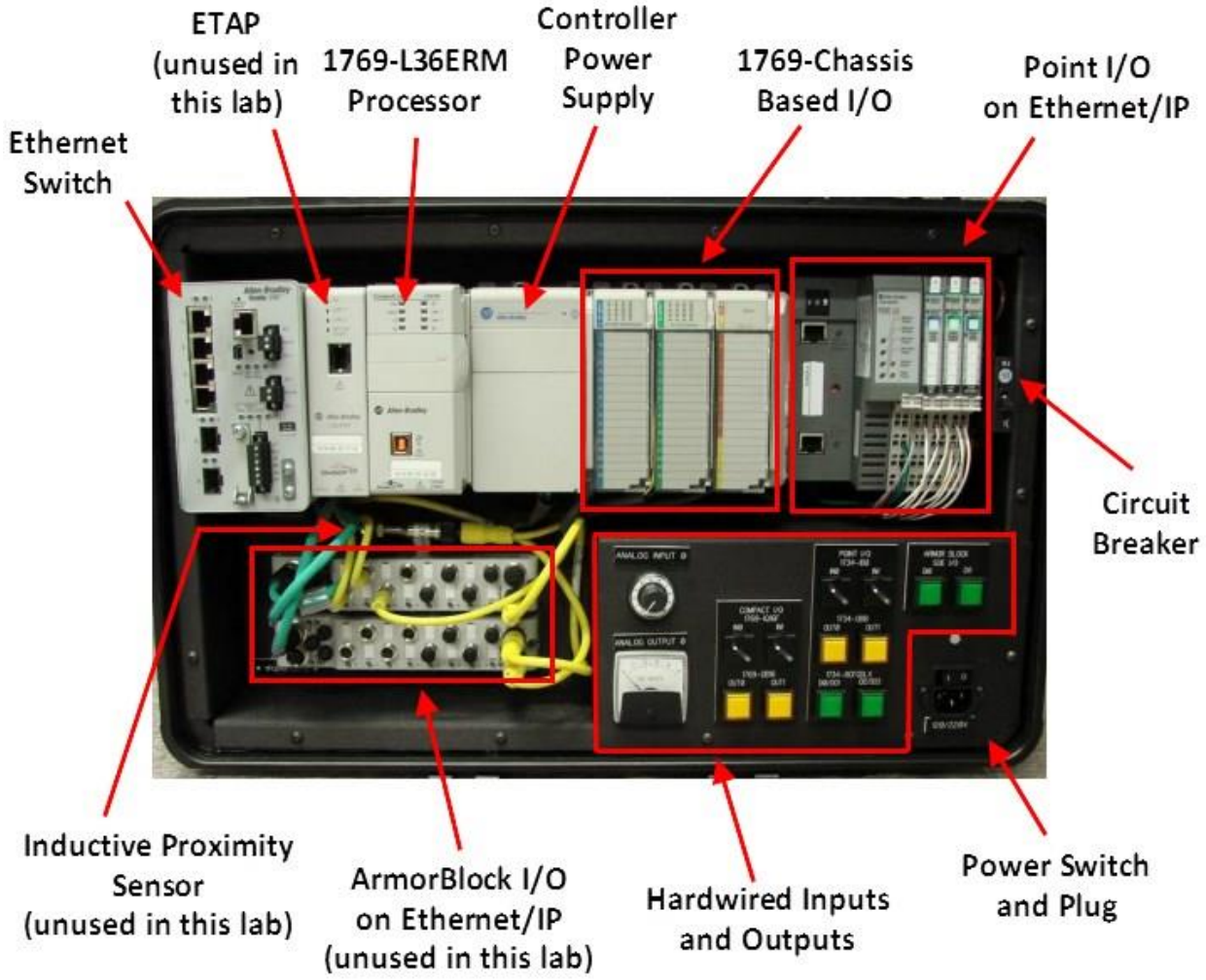
Ethernet Cable Routing:

Ethernet IP Addresses:

For the remainder of the *basic* lab

- Drive01 and Axis01 will refer to the Kinetix 5500 drive on the left side of the demo case.
- Drive02 and Axis02 will refer to the Kinetix 5500 on the right side of the demo case.
- Drive03 and Axis03 will refer to the PowerFlex 527 AC drive to the left of the Kinetix 5500s

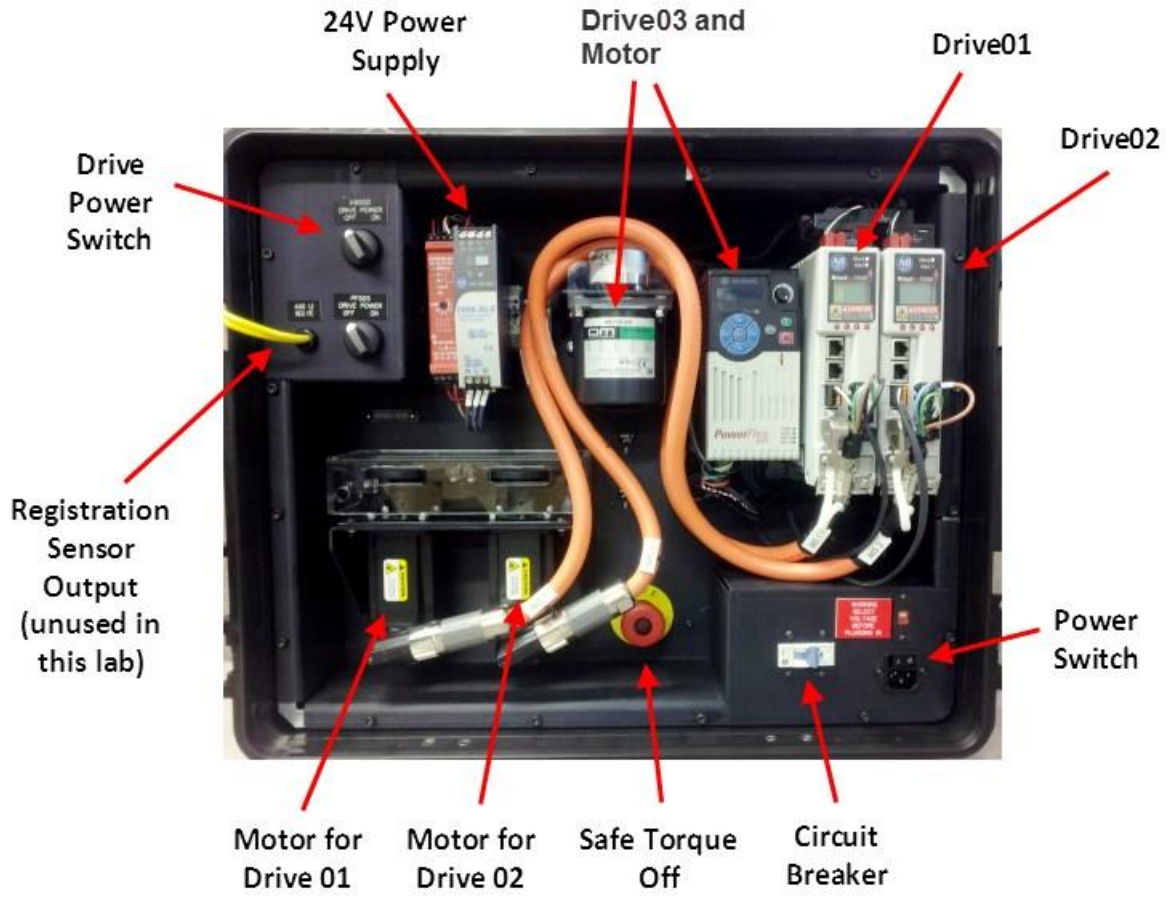
About the CompactLogix Demo



Use the image provided to locate these items on the demo and verify the lab setup:

- Verify the demo power switch labeled **"120/220V"** is **"on"**.
- Verify the circuit breaker is **"on"**.

About the Kinetix 5500 and PF527 3-Axis Demo



Use the image provided above to locate these items on the demo and verify the lab setup.

- Verify that the demo power switch is **"on"**.
- Verify that the circuit breaker is **"on"**.
- Verify that the switch labeled **"K5500 DRIVE POWER"** is **"on"**.
- Verify that the switch labeled **"PF525 DRIVE POWER"** is **"on"**. (Replaced with a PF527) ▪ Verify that the red mushroom button labeled **"SAFE OFF"** is pulled **"out"**.

Demonstration (Estimated Time 15 minutes)

Before starting the formal lab, let's begin with a brief demonstration showing the end result of the lab. During the demonstration, you will be able to control a complete 3 axis solution via an HMI. The demonstration will also allow you to independently control each axis, simulating what a real machine operator might need to do to clear a jam or manually control a portion of the machine. Following the demonstration you will move into the formal lab, where you will learn how to construct this solution with

detailed step-by-step directions. Along the way, the lab will highlight concepts important to Integrated Motion on Ethernet/IP. Let's begin…

Launch Studio 5000 and Open Application Files

1. Launch **Studio 5000**; double-click on the Studio 5000 desktop icon.

2. From the Open column, choose the Existing Project icon…

The Open Project window appears.

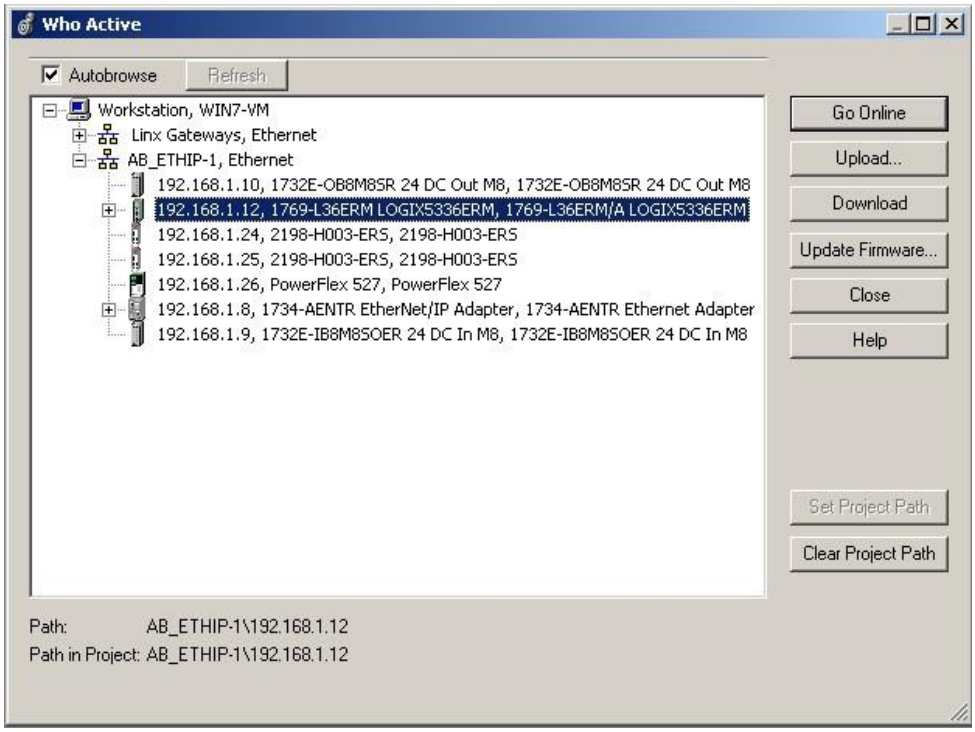
3. Browse to the folder **Lab Files** on the desktop and open **Integ_Motion_K5500_PF527_Complete.ACD**.

   Logix Designer opens.

4. Select Who Active from the Communications menu.

   The Who Active window appears.

5. Drill down through the **AB_ETHIP-1** driver and select the device at **192.168.1.12**, the CompactLogix processor.



If the path is not already set in the controller, click **Set Project Path**.

Note: If your control hardware is different than recommended or if you have any questions about the hardware, consult with your lab instructor.

6. Open the door on the front of the processor to reveal the Secure Digital card and the operation mode switch. Verify that the switch is in the **REM** (remote) position. Ensure that the **Kinetix 5500** and **PowerFlex 527** drives are each **"powered up"**.

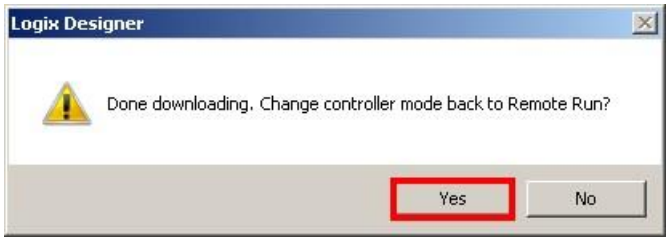
7. Click the **Download** button in the Who Active window.

   The Download window opens.

8. Click the **Download** button to send the program to the controller.

9. When downloading completes, place the controller into **Run Mode**.

   Method 1:



   Click **Yes**.

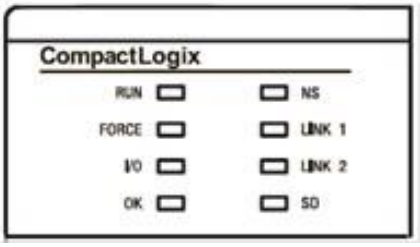
   Method 2:

Go to

and select **Run Mode**

Click **Yes**.

10. Verify that your controller is **"communicating"**…



- **OK** indicator should be **solid green**.
- **LINK1** indicator should be **flashing green**, indicating network activity
- **LINK2** indicator should be **flashing green**, indicating network activity

11. Verify that the Kinetix 5500 drives are "ready"…
    - **Module** light (7) should be **solid green**.
    - **Network** light (8) should be **solid green**.
    - **Port** status indicators (4 and 5) may be **flashing green**, indicating network traffic.

    Note: With the PF527 Beta units, the LEDs are not behaving properly. This issue has already been fixed. If your drive says STOPPED you are successfully connected.

12. The Kinetix 5500 and PF527 drives should read **"STOPPED"** in all capital letters across their display screen.

    It may take up to one minute for the drives to reach these states. If any of the above steps did not work as described, please consult your lab instructor.

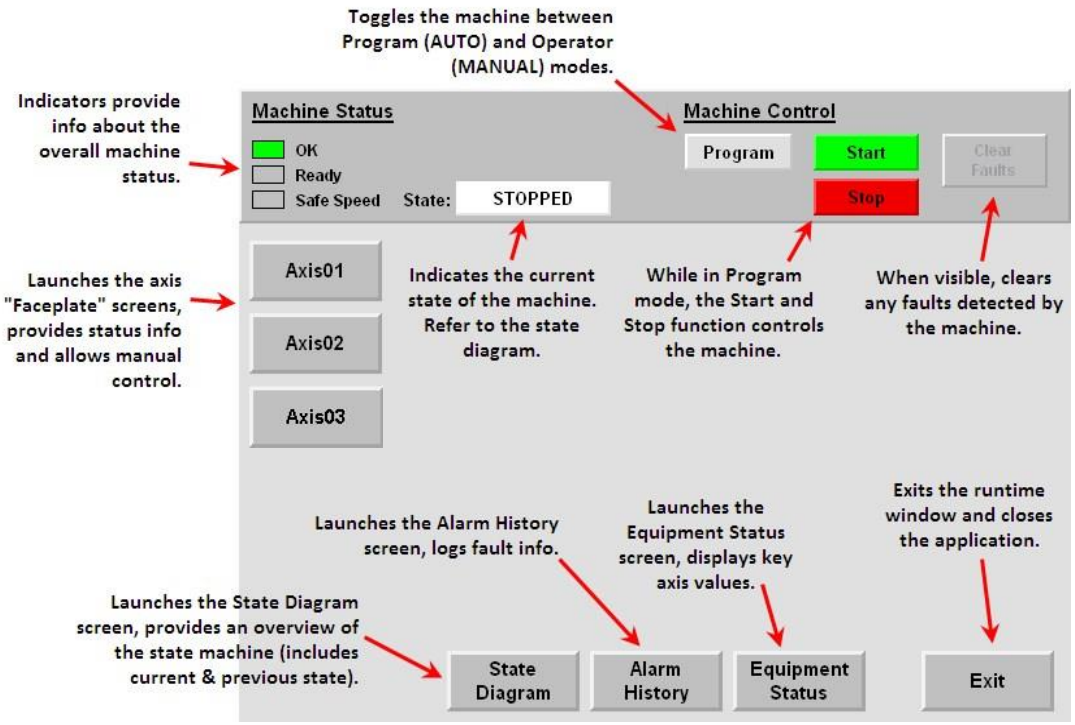
Open and Run the HMI Application

1. **Minimize** Logix Designer so that the HMI screen on the desktop can be seen.

2. Click **Start The Lab** on the warning screen of the HMI to load the

   Startup screen. 3. The **Startup** screen should initially be

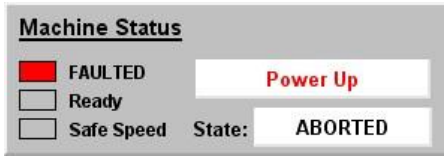
   displayed…


Toggles the machine between Program (AUTO) and Operator (MANUAL) modes.
Indicators provide info about the overall machine status.
Machine Status
OK
Ready
Safe Speed
State:
STOPPED
Machine Control
Program
Start
Stop
Clear Faults
Launches the axis "Faceplate" screens, provides status info and allows manual control.
Axis01
Axis02
Axis03
Indicates the current state of the machine. Refer to the state diagram.
While in Program mode, the Start and Stop function controls the machine.
When visible, clears any faults detected by the machine.
Launches the Alarm History screen, logs fault info.
Launches the Equipment Status screen, displays key axis values.
Exits the runtime window and closes the application.
Launches the State Diagram screen, provides an overview of the state machine (includes current & previous state).
State Diagram
Alarm History
Equipment Status
Exit

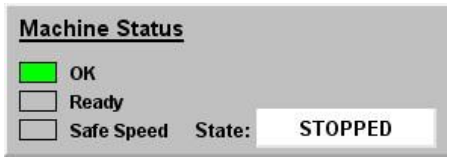
The **Startup** screen provides Machine Status and Control, plus it allows navigation to all other screens. Take a moment to familiarize yourself with the Startup screen before moving on to the next section. It may look a little different, depending on the status of the machine.

Start and Stop the Machine

1. If the machine is currently in the **ABORTED** state…



… press **Clear Faults**.



After a few moments the machine should transition to the **STOPPED** state.

2. Press **Program/Operator** button until **Program** is displayed. Program (AUTO) is now the active control mode.

3. Press **Start**.

The required axes enable and begin operating according to the Logix Designer program.

You should see that all three axes begin to rotate. If you notice closely, you might be able to tell that Axis02 appears to be following Axis01, but at approximately half speed. The program in Logix

Designer is gearing Axis02 to Axis01 at a 2:1 (Master : Slave) gear ratio, while Axis03 jogs at a constant speed using a Motion Drive Start instruction.

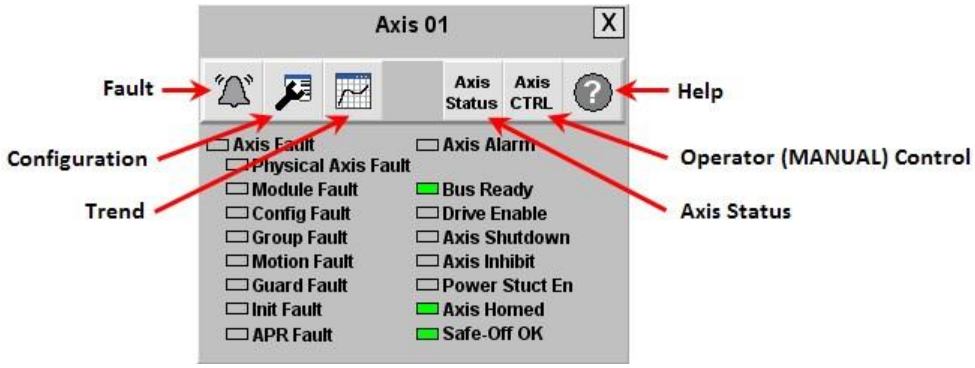
4. Press **Stop**.

   The motion system stops.

CIP Motion Axis Faceplate Manual Control

1. Press the

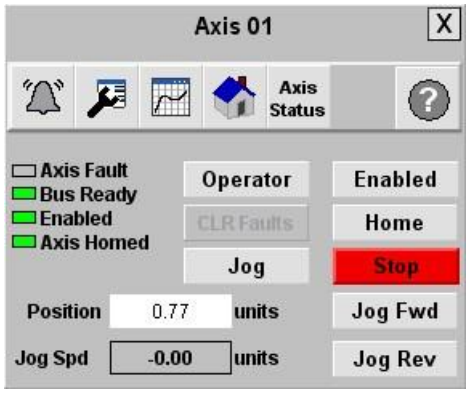
button from the **Startup** screen to launch the faceplate…



   The CIP Motion Axis faceplate provides axis status information, fault information, and trending data. The faceplate also includes the ability to manually control the axis.

   Press the

button…



2.

This picture shows the drive enabled, however your screen may differ when you first load the faceplate. From the **Axis CTRL** display, you have the ability to enable, disable, home, move, jog, and clear an axis fault.

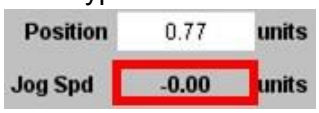
3. If **Program** is currently displayed, press the **Program/Operator** button until **Operator** is displayed. Operator (MANUAL) mode is now the active control mode.

4. Press the **Enabled/Disabled** button until **Enabled** is displayed.

   The axis should now be enabled and the **Enabled** indicator light should now be **solid green**.

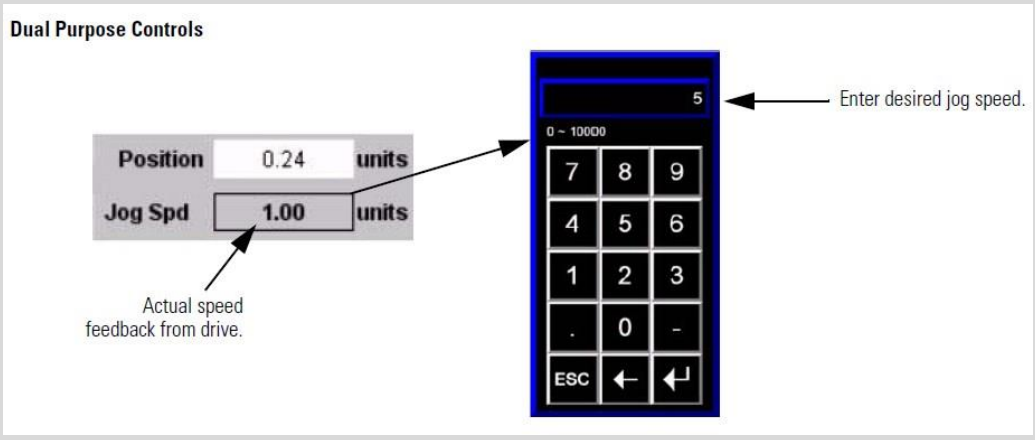
5. Press the **Jog/Move** button to select between the 2 types of manual control.

6. To set to jog speed for example, click on the corresponding **Jog Spd** display box to launch the keypad…



   The units for both Jog and Move manual control are **'revs'** and **'revs/sec'**

Note: Some of the numbers shown are both indicators and keypad input buttons. For example, the Jog Spd indicator displays the actual speed feedback of the drive, not the desired jog speed. However, by clicking the indicator you launch the keypad input object where you can enter the desired jog speed.



7. Take a few minutes to manually control the axis by executing a **Jog** command. To move the axis, press and hold the **Jog Fwd** or **Jog Rev** buttons.

8. This concludes the demonstration.
   - When you are finished manually controlling the axes, be sure to **Stop** and **Disable** all of the drives.
   - Maximize Logix Designer, and go offline with the current file by selecting **Go Offline** from the **Communications** menu.

Lab 1: Basic Configuration of an Integrated Motion System (Estimated Time 20 Minutes)

In this lab, we will introduce you to the **CompactLogix with Integrated Motion on EtherNet/IP** product family by performing the following:

- Creating a project by utilizing the *'Drives & Motion Accelerator Toolkit'*.
- Learn about the core concepts and benefits of Integrated Motion on EtherNet/IP.
- Configure your motion hardware including controller and drives.

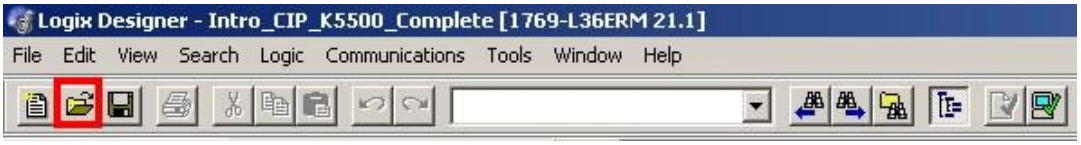
Note: Much of the information and detailed steps provided in this lab can also be found in the *'CIP Motion Configuration and*

Startup User Manual' (MOTION-UM003-EN-P) available via Literature Library. More information about the programming techniques used here can be found in the publication *'Drives and Motion Accelerator Toolkit Quick Start'* (IASIMP-QS019-EN-P) available via Literature Library.

Open the Application File

1. Return to Logix Designer.

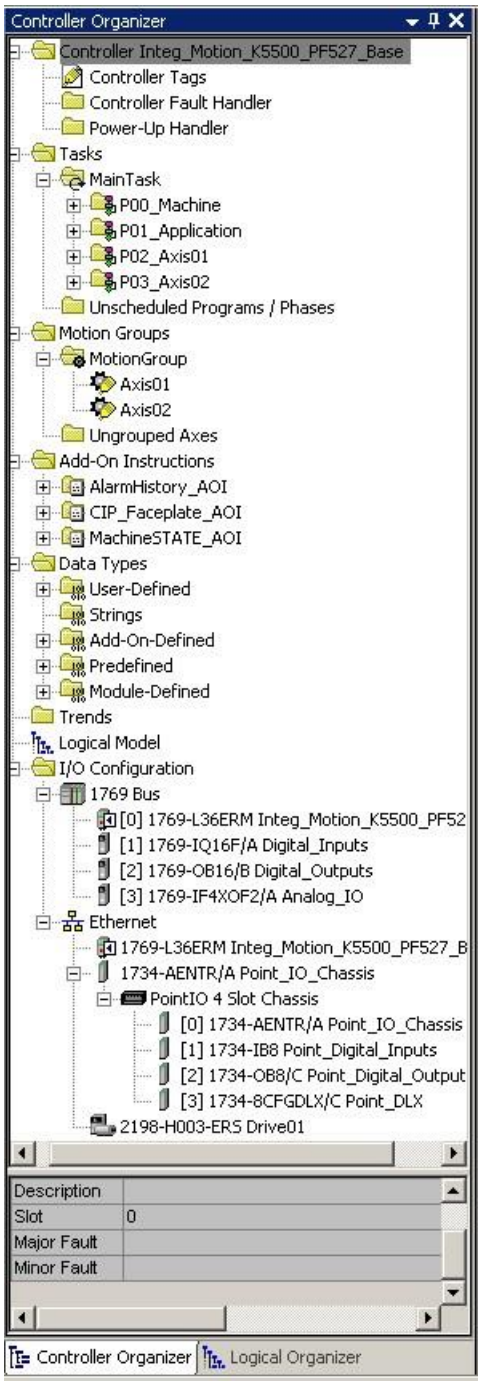
2. From the Tool Bar menu, choose the Open icon…



   The Open window appears. You do not need to save changes to your existing file.

3. Browse to the folder **Lab Files** on the desktop and open file **Integ_Motion_K5500_PF527_Base.ACD**. When you open the logic file, the Controller Organizer appears on the left side of the Studio 5000 window.

   The *'Drives & Motion Accelerator Toolkit'* is a modular programming structure that was used to create this sample logic file. The sets of tools included with the *'Drives and Motion Accelerator Toolkit'* provide pre-configured example logic that can be customized to meet the needs of motion applications – a "quick start" to programming your drives and motion system.


Controller Organizer
Controller Integ_Motion_K5500_PF527_Base
Controller Tags
Controller Fault Handler
Power-Up Handler
Tasks
MainTask
P00_Machine
P01_Application
P02_Axis01
P03_Axis02
Unscheduled Programs / Phases
Motion Groups
MotionGroup
Axis01
Axis02
Ungrouped Axes
Add-On Instructions
AlarmHistory_AOI
CIP_Faceplate_AOI
MachineSTATE_AOI
Data Types
User-Defined
Strings
Add-On-Defined
Predefined
Module-Defined
Trends
Logical Model
I/O Configuration
1769 Bus
[0] 1769-L36ERM Integ_Motion_K5500_PF52
[1] 1769-IQ16F/A Digital_Inputs
[2] 1769-OB16/B Digital_Outputs
[3] 1769-IF4XOF2/A Analog_IO
Ethernet
1769-L36ERM Integ_Motion_K5500_PF527_B
1734-AENTR/A Point_IO_Chassis
PointIO 4 Slot Chassis
[0] 1734-AENTR/A Point_IO_Chassis
[1] 1734-IB8 Point_Digital_Inputs
[2] 1734-OB8/C Point_Digital_Output
[3] 1734-8CFGDLX/C Point_DLX
2198-H003-ERS Drive01
Description
Slot
0
Major Fault
Minor Fault
Controller Organizer
Logical Organizer

The Controller Organizer is a graphical representation of the contents of your controller project. This display consists of a tree of folders that contain all of the information about the programs and data in the current controller project. The default main folders in this tree are:

Controller Project Name – Contains the controller-scoped tags, controller fault handler, and the power up handler. ***Tasks*** – Tasks are shown in this folder. Each task contains its own programs with routines and program scoped tags. The routines can be: ladder diagrams, sequential function chart, function block diagram, and/or structured text. ***Motion Groups*** – Underneath the Motion Groups folder, you will find one group of axes which contains individual axes as well as coordinate systems. In addition, you will find Ungrouped Axes, which are axes that have yet to be assigned to any particular group. You can assign these axes to the motion group via the Axis Assignment tab of the Motion Group Properties window.

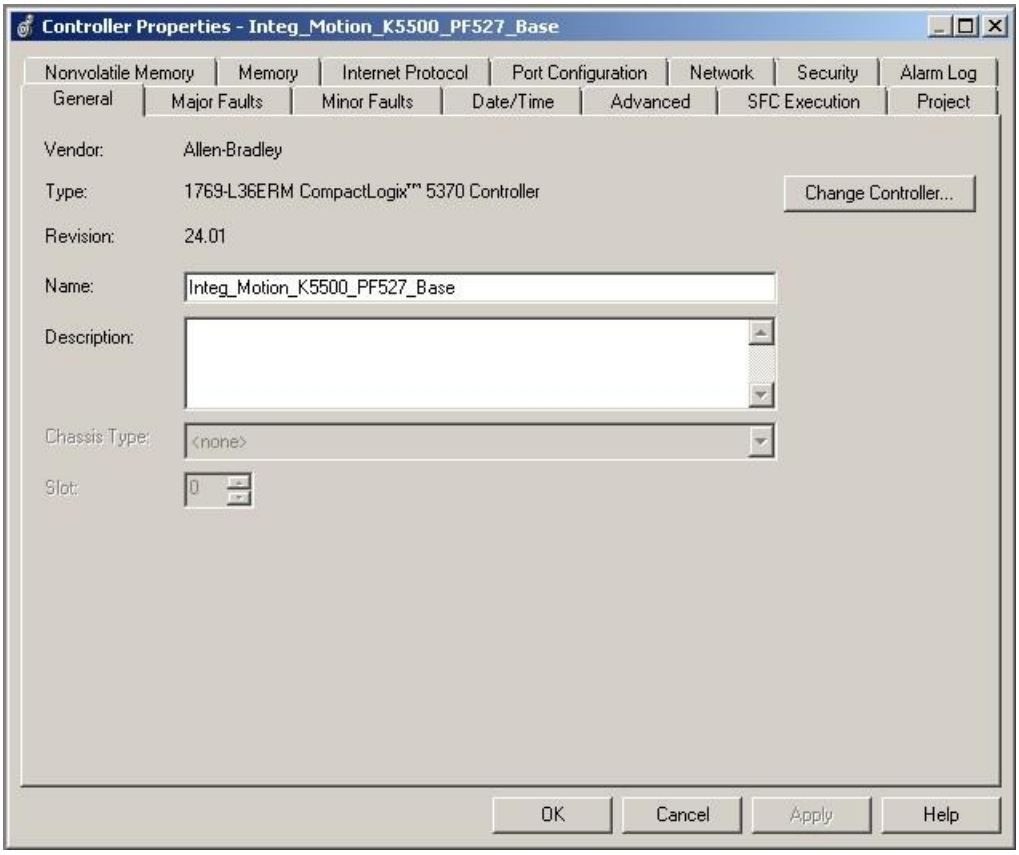
Add-On Instructions – Add-On Instructions are instructions that you define, or they can be provided to you by someone else. Once defined in the project, they are similar to the built-in instructions already in the Logix controllers. An Add-On Instruction allows you to encapsulate your most commonly used logic as sets of instructions. They are useful for commonly used instructions in your projects and to promote consistency across the projects.

Trends – Trends are shown in this folder.

Data Types – Shows predefined and user-defined data types. User-defined data is created in this folder. ***I/O Configuration*** – Contains the information about the hardware configuration of this controller project. It holds a hierarchy of modules with which the controller is configured to communicate.

4. Select Controller Properties from the Edit menu.

   The Controller Properties window opens…

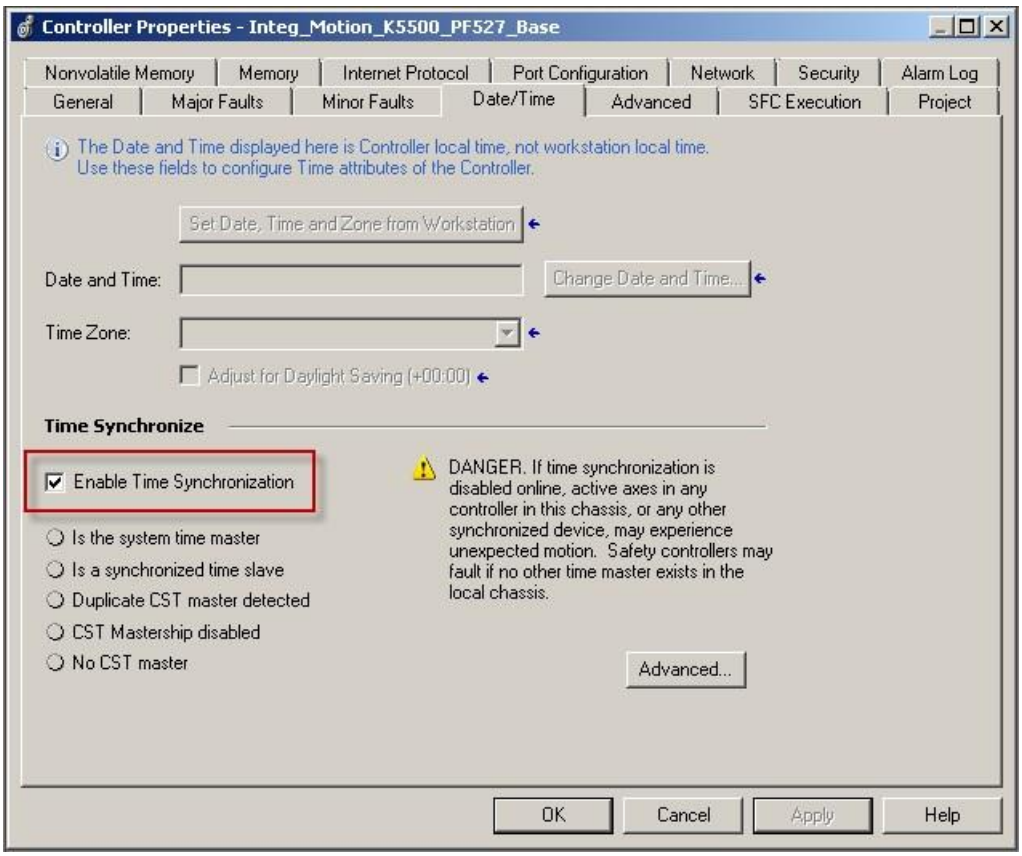


5. Select the **General** Tab.

   Notice that the controller type has been selected for you already. With the 1769-L36ERM CompactLogix controller that we are using in this hardware setup, the slot and chassis type cannot be changed by the user.

   Note: If your controller hardware is different than specified, please consult with your lab instructor to make the appropriate changes.

6. Select the **Date/Time** tab…



Verify that the selection box 'Enable Time Synchronization' is checked.

Controllers that currently support the PF527 include 1756-L7x/ L7xS and CompactLogix PAC ERM controllers with v24 firmware.

'Enable Time Synchronization' differs from *'Make this controller the Coordinated System Time master'* that was implemented in previous versions of RSLogix 5000. 'Enable Time Synchronization' establishes the module's ability to participate in time synchronization, which is a fundamental requirement of CIP Motion.

CIP Motion doesn't rely on a rigid, scheduled network to create determinism. Instead, CIP Motion delivers the data and timestamp for execution as a part of a standard Ethernet packet. This allows motion devices to plan and follow positioning path information according to a pre-determined execution plan. **The controller, communication module, and all of the motion devices require time synchronization for CIP Motion to function**.

The mechanism that provides time synchronization on EtherNet/IP is referred to as CIP Sync. CIP Sync is based on the IEEE-1588 Precision Time Protocol (PTP) standard, which details synchronizing time for devices connected in a network.

The sole system time master is referred to as the Grandmaster and is determined by a strict arbitration process. By default the Grandmaster is both PTP / Coordinated System Time (CST) master and typically will be a viable communication module or processor. The settings on the 'Advanced' window (Date/Time tab) can allow the processor to win the arbitration over the other processors and/or communication modules connected to it.

The following example illustrates the Grandmaster / Master / Slave relationship for a ControlLogix chassis and it's connected I/O; the same would hold true for eligible CompactLogix controllers.

Note: In systems with multiple processors, all controllers need to have time synchronization enabled if they are to use CST / PTP time.

The System Time timestamp is a 64-bit (LINT) value that represents the number of nanoseconds or microseconds starting from January 1, 1970 at 12:00 am.

7. Click **OK** to close the Controller Properties window.

Hardware and Network Considerations

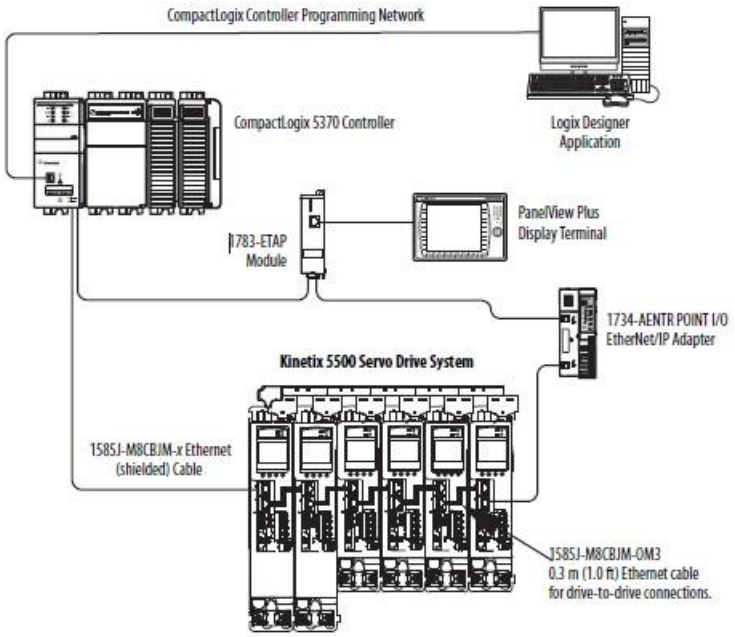
Before we continue on with the lab, let's discuss some of the hardware and network options that are available with Integrated Motion on EtherNet/IP.

Network Topology

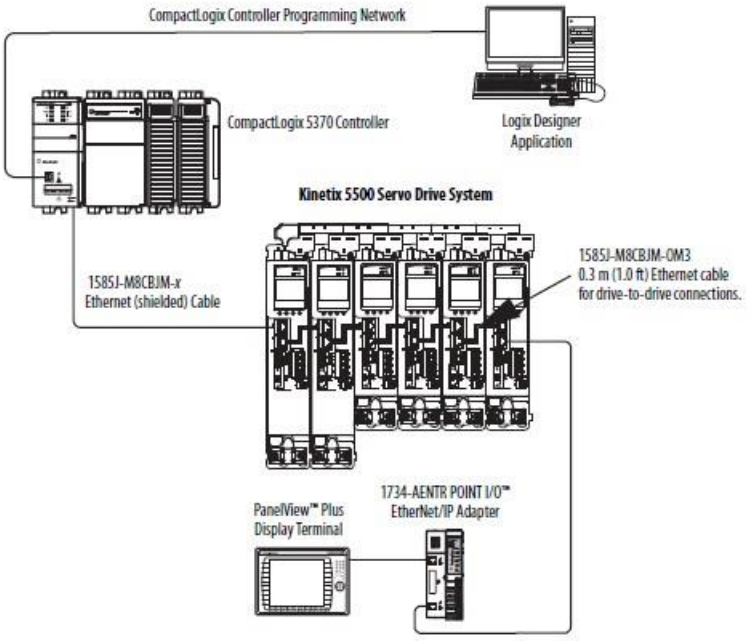
Integrated Motion on EtherNet/IP allows for multiple network topologies, providing the flexibility necessary to meet even the most demanding applications. Listed below are 3 of the more popular network topologies.

Note: These diagrams were sourced from the *'CompactLogix 5370 Controllers User Manual'* (1769-UM021 –EN-P). More network topologies are shown in the *'CIP Motion Popular Configuration Drawings'* (IASIMP-QR019 –EN-P) available via the Rockwell Automation Literature Library.

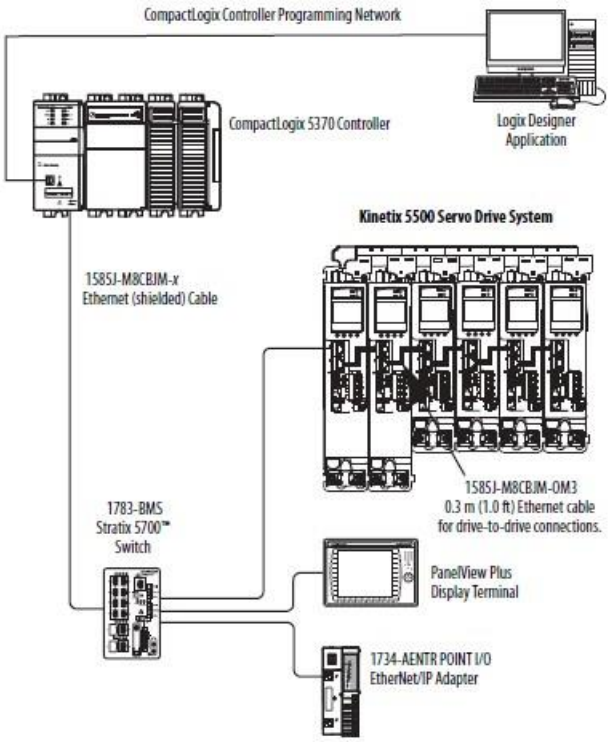
Device Level Ring with Integrated Motion


CompactLogix Controller Programming Network
CompactLogix 5370 Controller
Logix Designer Application
1783-ETAP Module
PanelView Plus Display Terminal
1734-AENTR POINT I/O EtherNet/IP Adapter
Kinetix 5500 Servo Drive System
1585J-M8CBJM-x Ethernet (shielded) Cable
1585J-M8CBJM-OM3 0.3 m (1.0 ft) Ethernet cable for drive-to-drive connections.

Linear with Integrated Motion


CompactLogix Controller Programming Network
CompactLogix 5370 Controller
Logix Designer Application
Kinetix 5500 Servo Drive System
1585J-M8CBJM-x Ethernet (shielded) Cable
1585J-M8CBJM-OM3 0.3 m (1.0 ft) Ethernet cable for drive-to-drive connections.
1734-AENTR POINT I/O™ EtherNet/IP Adapter
PanelView™ Plus Display Terminal

Star with Integrated Motion



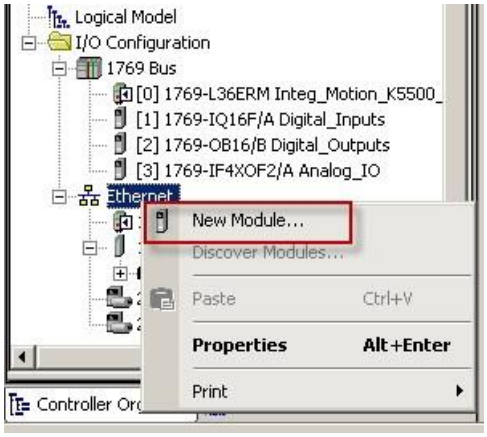
Add Your Drive Hardware

In this section you add the following drive hardware to the I/O configuration:

- **Drive03** (PowerFlex 527)

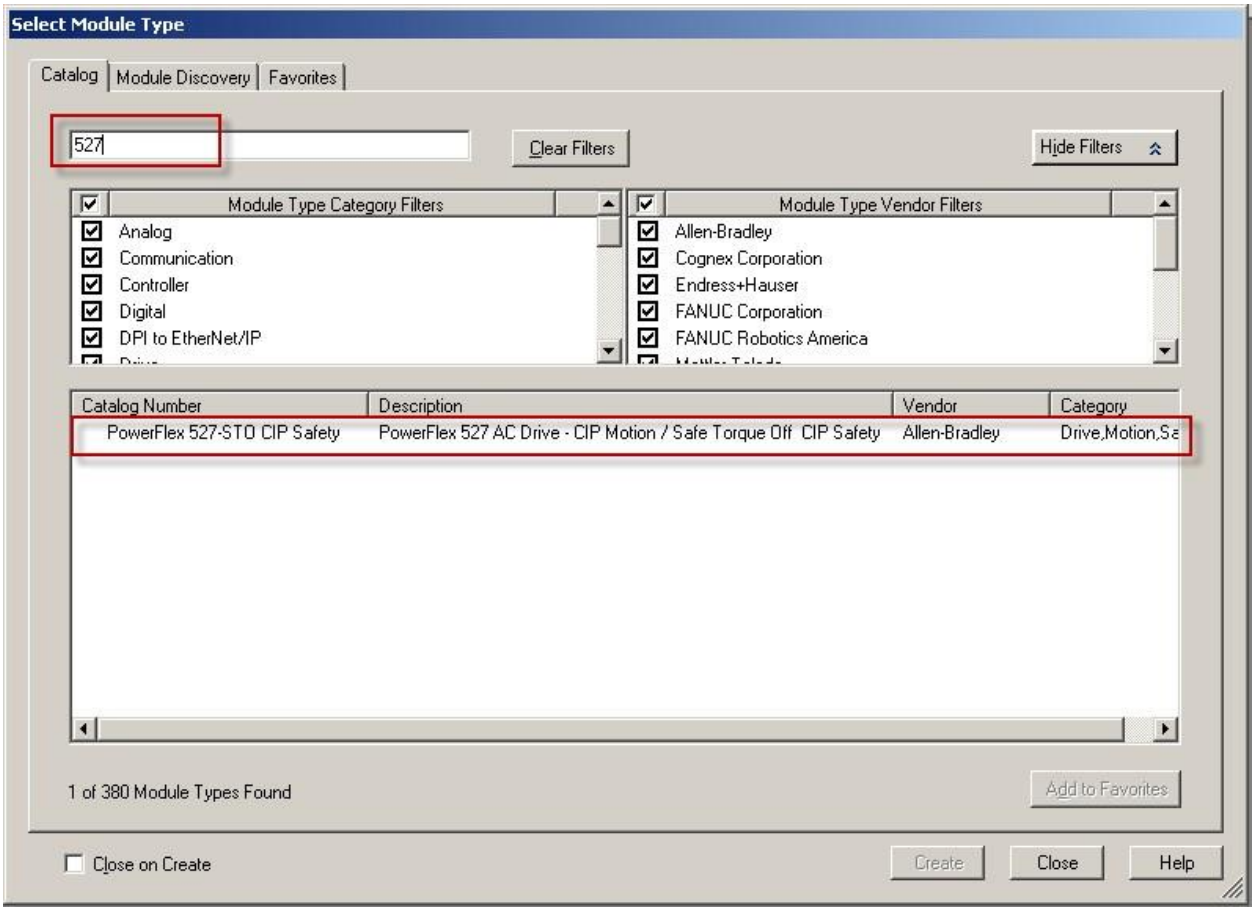
Note: **Drive01 and Drive02** (Kinetix 5500s) have already been pre-configured for you.

1. Right-click on the Ethernet network icon and select New Module…

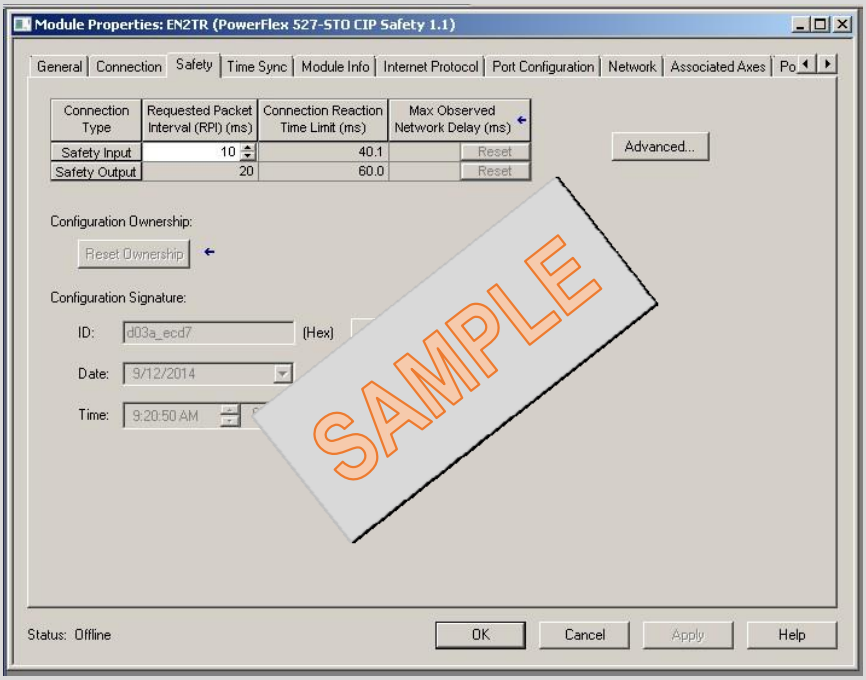


The Select Module Type window opens.

2. In the search box, type ***'527'*** and watch as the list repopulates. Select the catalog number **PowerFlex 527-STO CIP Safety**.



TIP: The PF527 has embedded dual EtherNet/IP and Integrated Safety all included! The only option card required for the PF527 is an encoder card. When in the I/O tree with a safety controller, the Safety tab is available to configure your safety connection. See below.

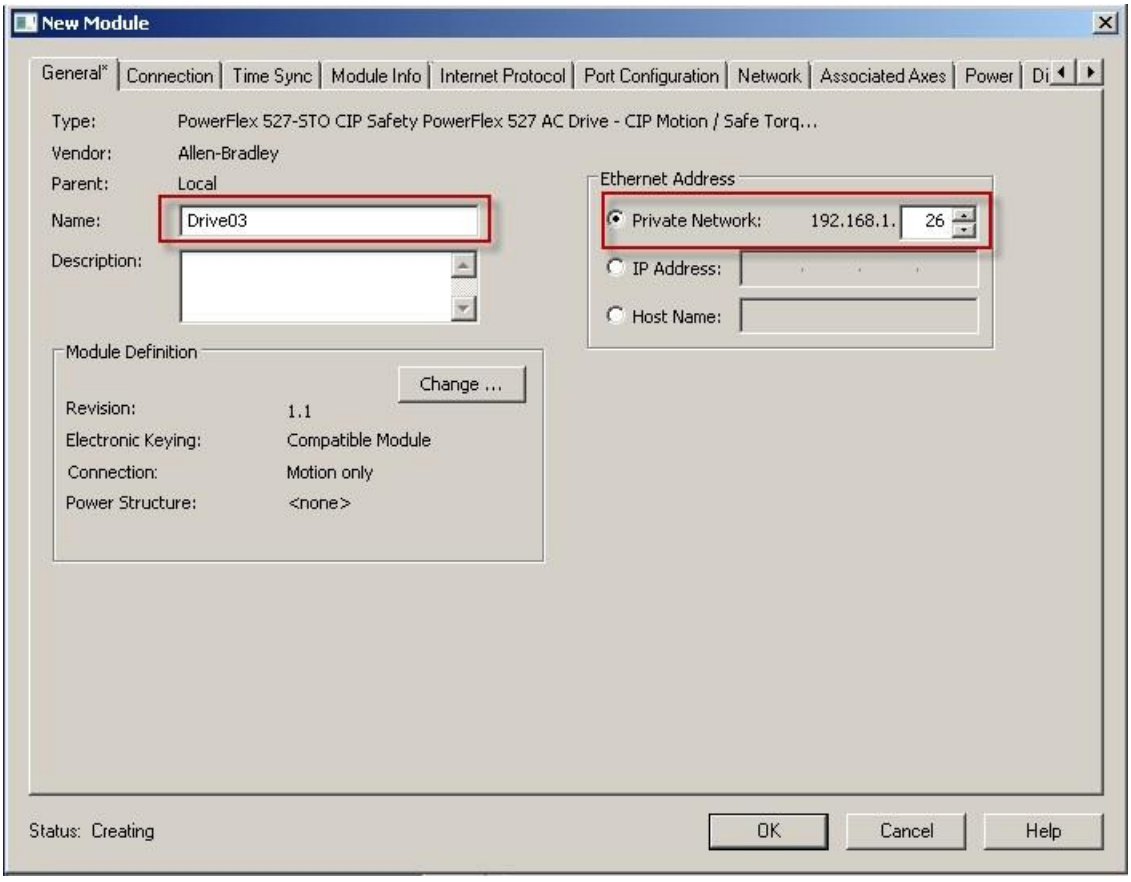


See the Advanced Safety Lab in the Appendix for more information about Integrated Network Safety.

3. Click **Create**.

   The New Module window will appear.

4. On the **General** tab…



(1) Type **'Drive03'** in the Name field.

(2) Select **Private Network** and set the Ethernet address to 192.168.1.**26**.

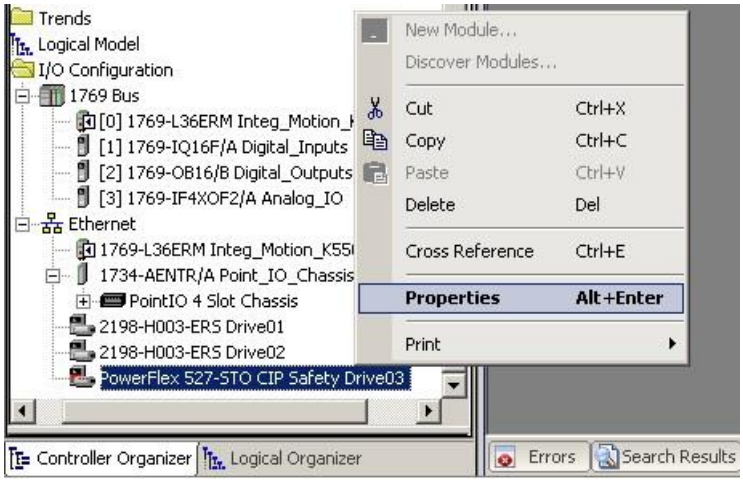
(3) Click **OK**.

> **TIP:** The PF527 IP address can be configured from the HIM of the drive OR you can use DHCP to assign an IP address. On the HIM navigate to Settings -> Network -> Static to change the static IP address or Settings -> Network -> DHCP to setup DHCP.

5. If the Select Module Type window is still open, press **Close**.

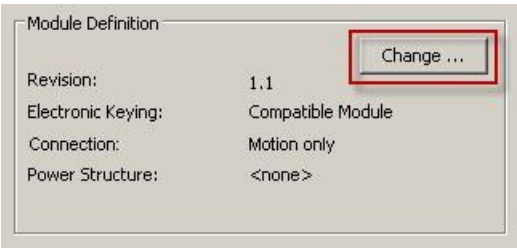
   The drive that you just added should now appear under the Ethernet network in your I/O configuration.

6. To complete the drive configuration, right-click on **Drive03** and select **Properties**…



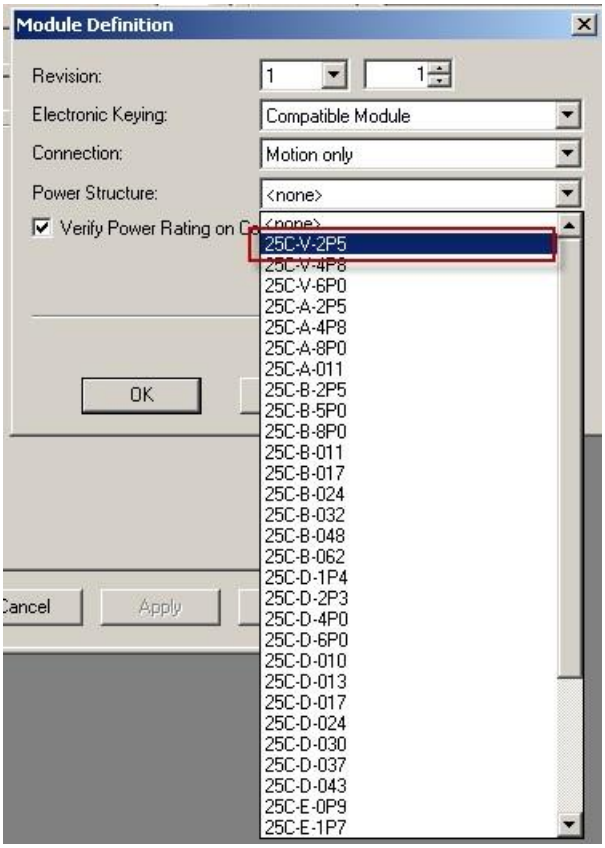
The Module Properties window opens.

7. From the General tab click the **Change** button.

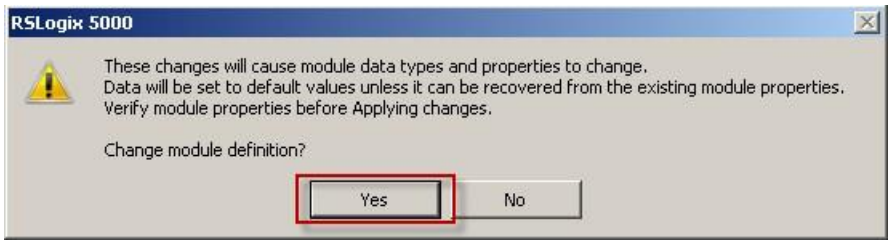


8. Select **25C-V-2P5** from the Power Structure drop down menu and click **OK**.

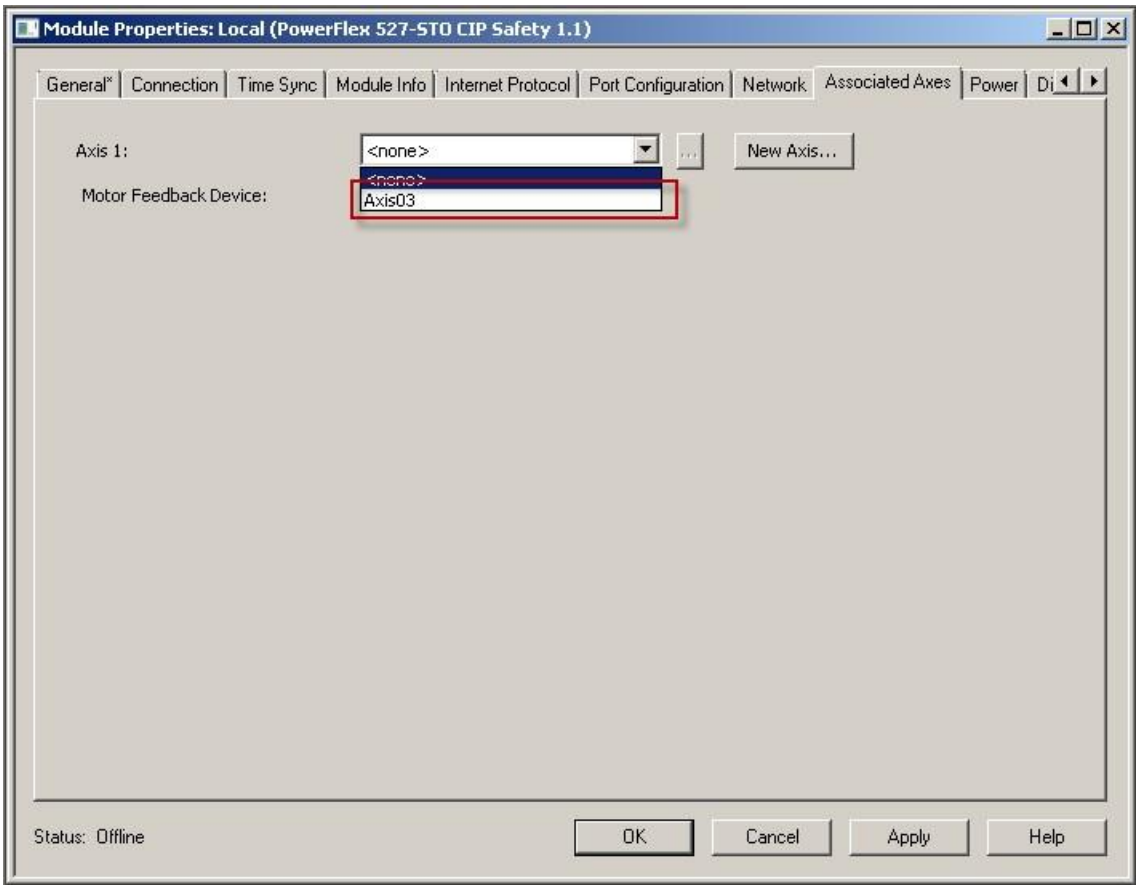
TIP: The PowerFlex 527 utilizes the same power core as the PowerFlex 525. Simply swap the control cores!



9. Click **Yes** to confirm module configuration change.


RSLogix 5000
These changes will cause module data types and properties to change.
Data will be set to default values unless it can be recovered from the existing module properties.
Verify module properties before Applying changes.
Change module definition?
Yes
No

10. Navigate to the **Associated Axes** tab…



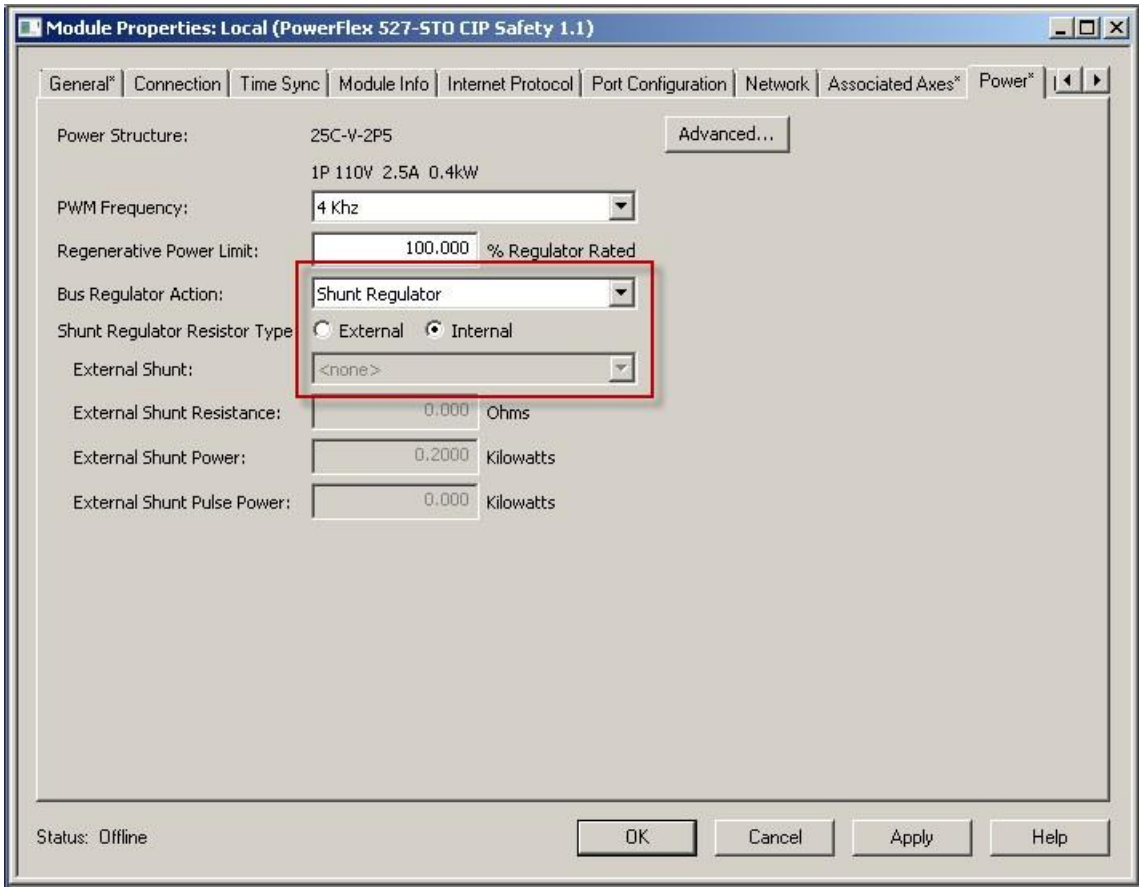
In the drop down menu for **Axis 1**, choose **'Axis03'**.

> **TIP:** Though this drive type does not support auxiliary feedback, other CIP Motion drives do. In these drives, Axis 1 and Axis 2 are both listed. The auxiliary feedback port on those drives can be used for load feedback of the primary axis if the axis has a Feedback Configuration of Load or Dual.
>
> Axis 2 is for a Feedback Only or "half" axis. Typically a Feedback Only axis will act as a master reference for electronic gearing applications.

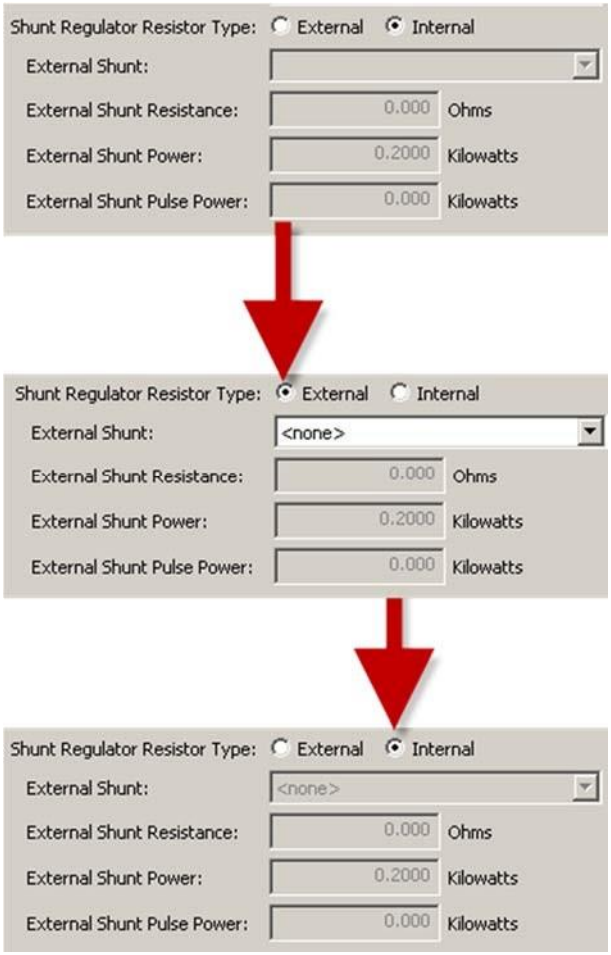
11. Navigate to the **Power** tab and verify the following:

    Bus Regulator Action: **Shunt Regulator**

    Shunt Regulator Resistor Type: **Internal**



12. Verify that External Shunt shows **<none>** even though it is grayed out. Do this by clicking the **External** radio button, then clicking the **Internal** radio button.


Shunt Regulator Resistor Type: External Internal
External Shunt:
External Shunt Resistance: 0.000 Ohms
External Shunt Power: 0.2000 Kilowatts
External Shunt Pulse Power: 0.000 Kilowatts
Shunt Regulator Resistor Type: External Internal
External Shunt: <none>
External Shunt Resistance: 0.000 Ohms
External Shunt Power: 0.2000 Kilowatts
External Shunt Pulse Power: 0.000 Kilowatts
Shunt Regulator Resistor Type: External Internal
External Shunt: <none>
External Shunt Resistance: 0.000 Ohms
External Shunt Power: 0.2000 Kilowatts
External Shunt Pulse Power: 0.000 Kilowatts

Click **Apply** then **OK** to close the Module Properties window for **Drive03**.

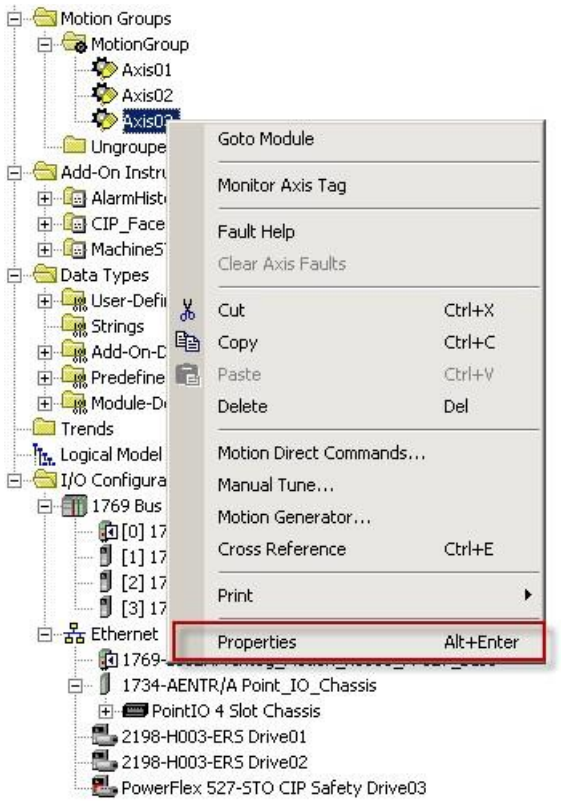
Configure Axis Properties

In this section you will configure the following Axis:

- **Axis03** (PowerFlex 527 and associated induction motor)

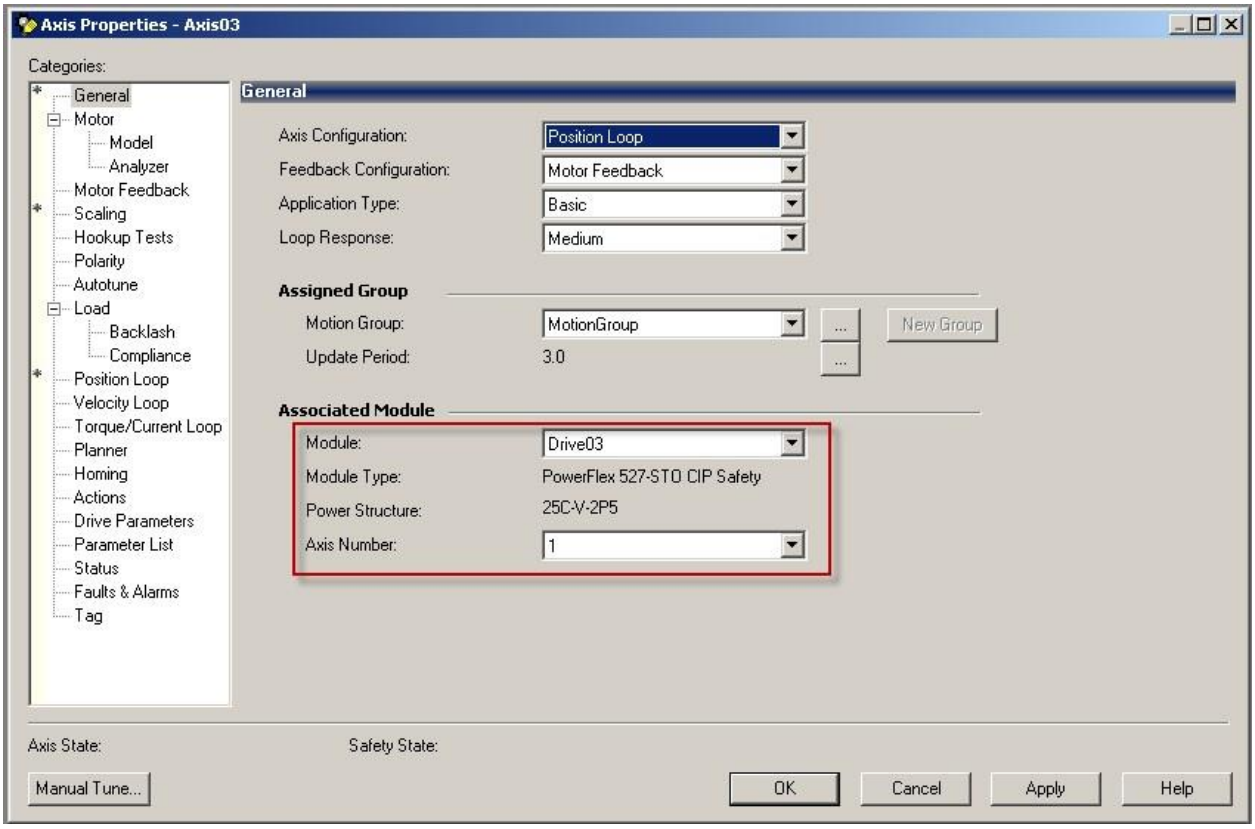
Note: **Axis01 and Axis02** (on the left) has already been pre-configured for you.

1. From the **Motion Groups > MotionGroup** folder in the Controller Organizer, right-click on **Axis03** and select **Properties**…



The Axis Properties window opens.

2. Notice on the **General** page that the PowerFlex 527 drive module you added in the previous section is assigned to this axis…



Do not close the Axis Properties window until instructed to do so.

There are three **Axis Configuration** options for the PowerFlex 527 drive:

Frequency Control, Position Loop, Velocity Loop

The Application Type and Loop Response are used to configure the axis to optimize the Autotune results. Use the description and table to determine the most appropriate configuration for your typical machine.

There are five different **Application Types** for Integrated Motion drives:

Custom – Advanced tuning, user selects Autotune parameters

Basic – Default tuning parameters

Tracking – Winding/unwinding, flying shear and web control applications

Point-to-Point – Pick & place, packaging, and cut to length applications

Constant Speed – Conveyors, line shaft, or crank applications

	Position Loop Bandwidth	Position Integrator Bandwidth	Velocity Loop Bandwidth	Velocity Integrator Bandwidth	Integrator Hold	Velocity Feedforward	Acceleration Feedforward
Custom	X		X				
Basic	X		X				
Tracking	X		X	X		X	X
Point-to-Point	X	X	X		X		
Constant Speed	X		X	X		X	

X - selected by default
When selected, Autotune will set the Velocity & Acceleration Feedforward to 100%

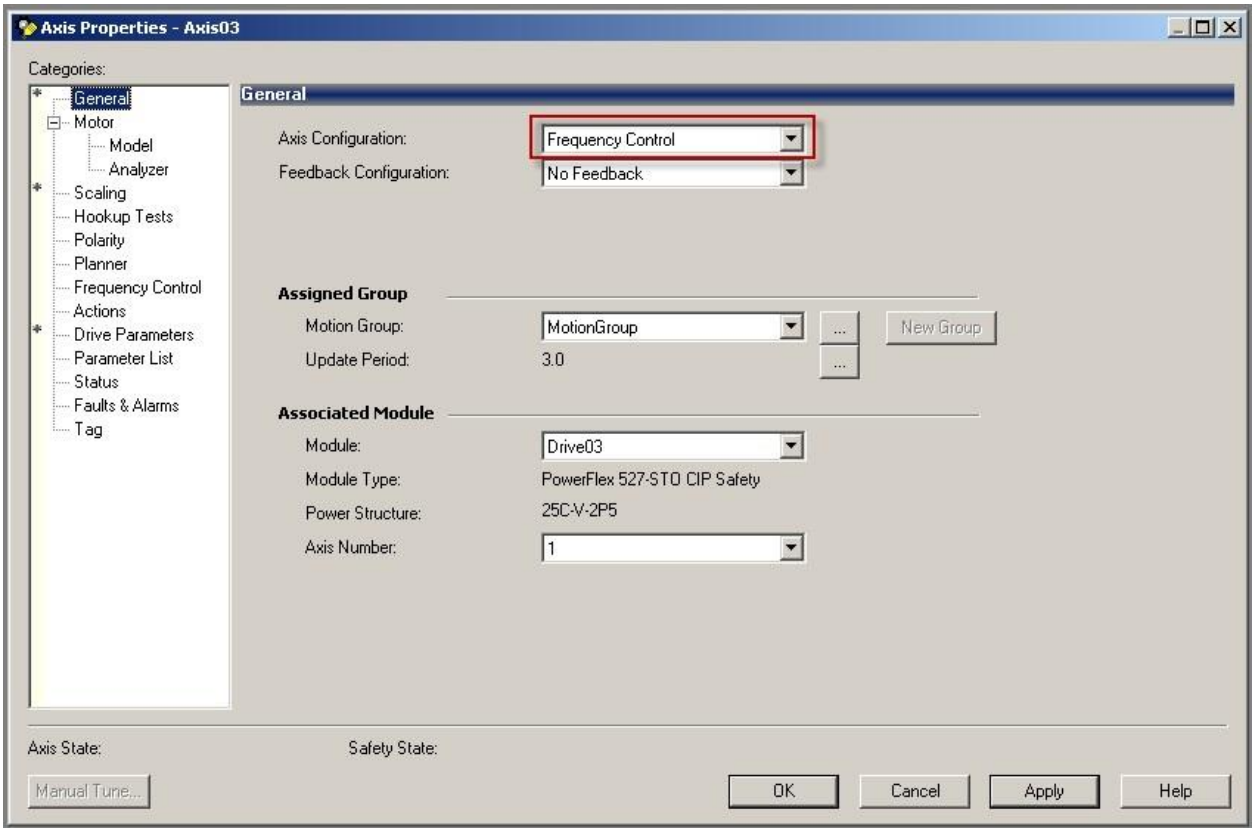
There are three **Loop Response** options:

Low – Damping Factor = 1.5

Medium – Damping Factor = 1.0

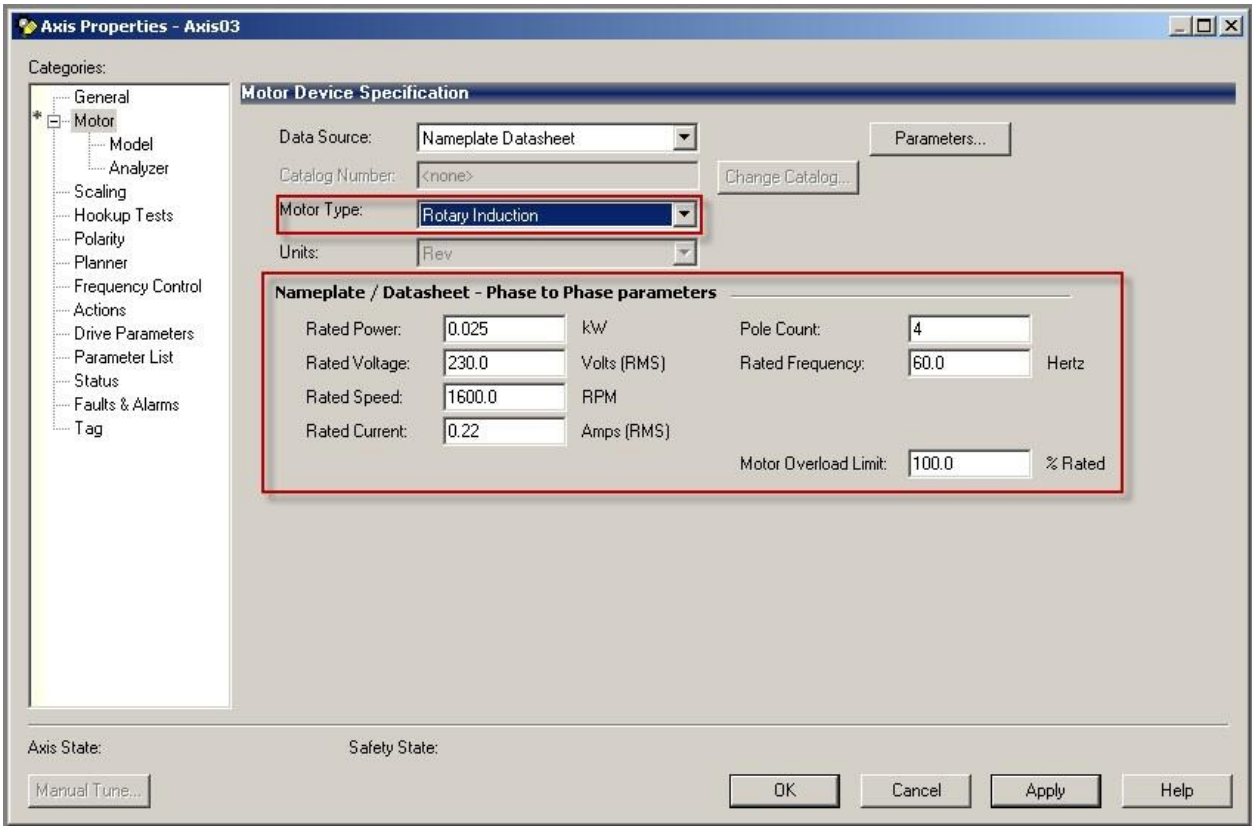
High – Damping Factor = 0.8

3. Select **Frequency Control** from the Axis Configuration drop down.



For the beta/early version of the PF527 in this lab, only Frequency Control has been tested. At release, there will be an encoder card available for closed loop Velocity and Position control for the PF 527 - Incremental A quad B with Z channel (z channel = marker pulse) encoder support.

4. Navigate to the **Motor** page…



(1) For the **Data Source**, select **'Nameplate Datasheet'**.

(2) For **Motor Type**; select **'Rotary Induction'**.

(3) Enter the following motor data:

Rated Power – **0.025 kW**

Rated Voltage – **230 Volts (RMS)**

Rated Speed – **1600 RPM**

Rated Current – **0.22 Amps**

Pole Count – **4**

Rated Frequency – **60 Hz**

Motor Overload Limit – **100% Rated**

Data Source options for K5500 and PF527:

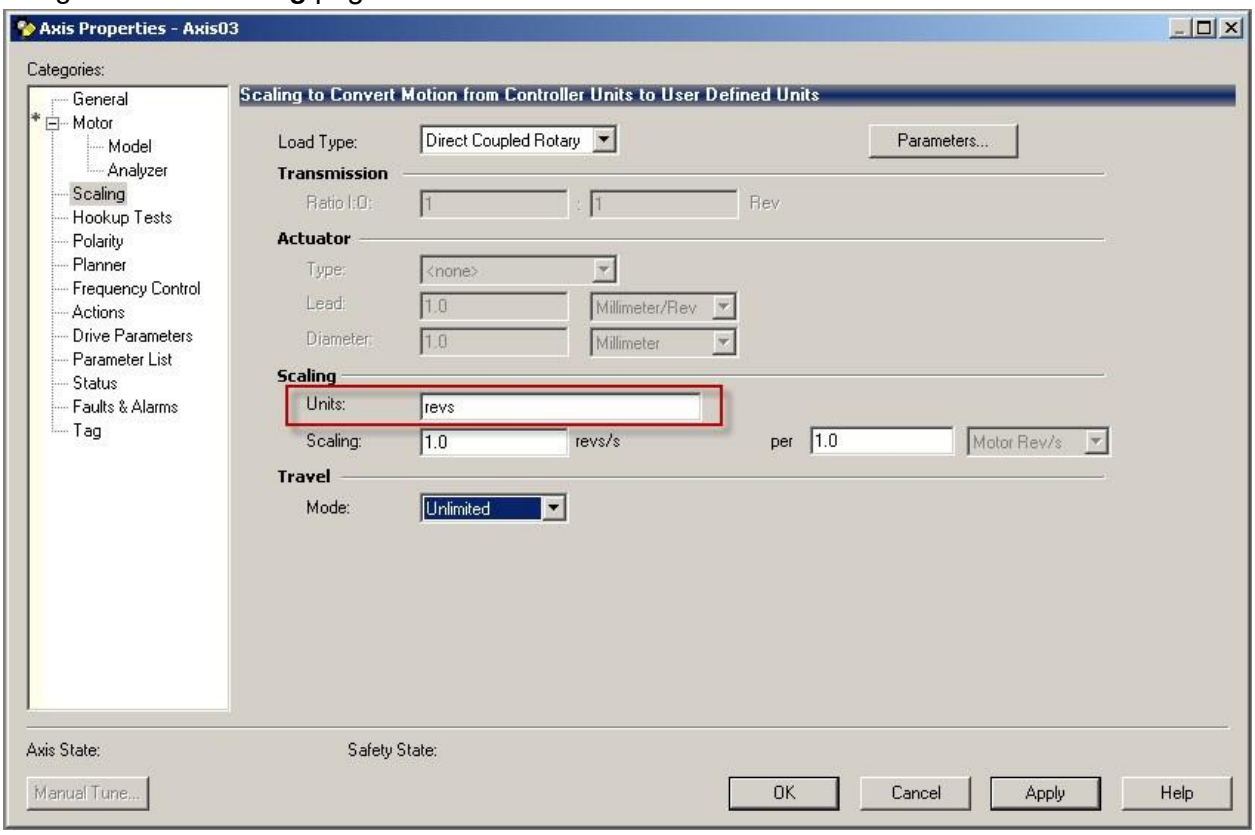
Nameplate Datasheet – Motor parameters are entered directly by the user. Optional for those users who have experience with servo motor data and wish to enter their own 3rd Party motor parameters.

Catalog Number – For K5500 motors where parameters are acquired from the Motion Database. Customers will generally employ an AB motor listed in the Motion Database.

Motor NV – Motor parameters are derived from non-volatile memory of a motor-mounted smart feedback device equipped with a serial interface. Applies to any Hiperface or EnDat based motor which is "preprogrammed" with Rockwell Automation formatted data.

Click **Apply** to save your changes.

5. Navigate to the **Scaling** page…

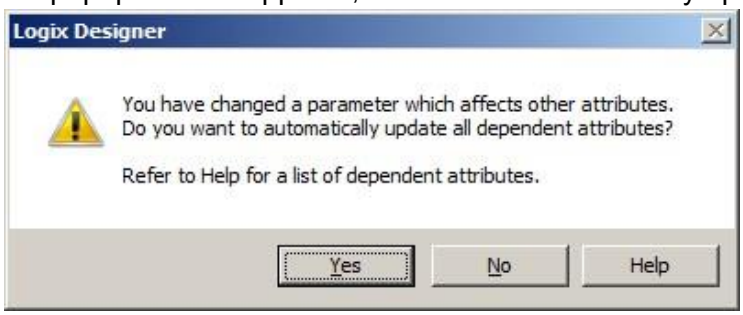


(1) Enter **'revs'** in the **Scaling Units** box. Leave the Scaling set to **1.0** revs per **1.0** Motor Rev.

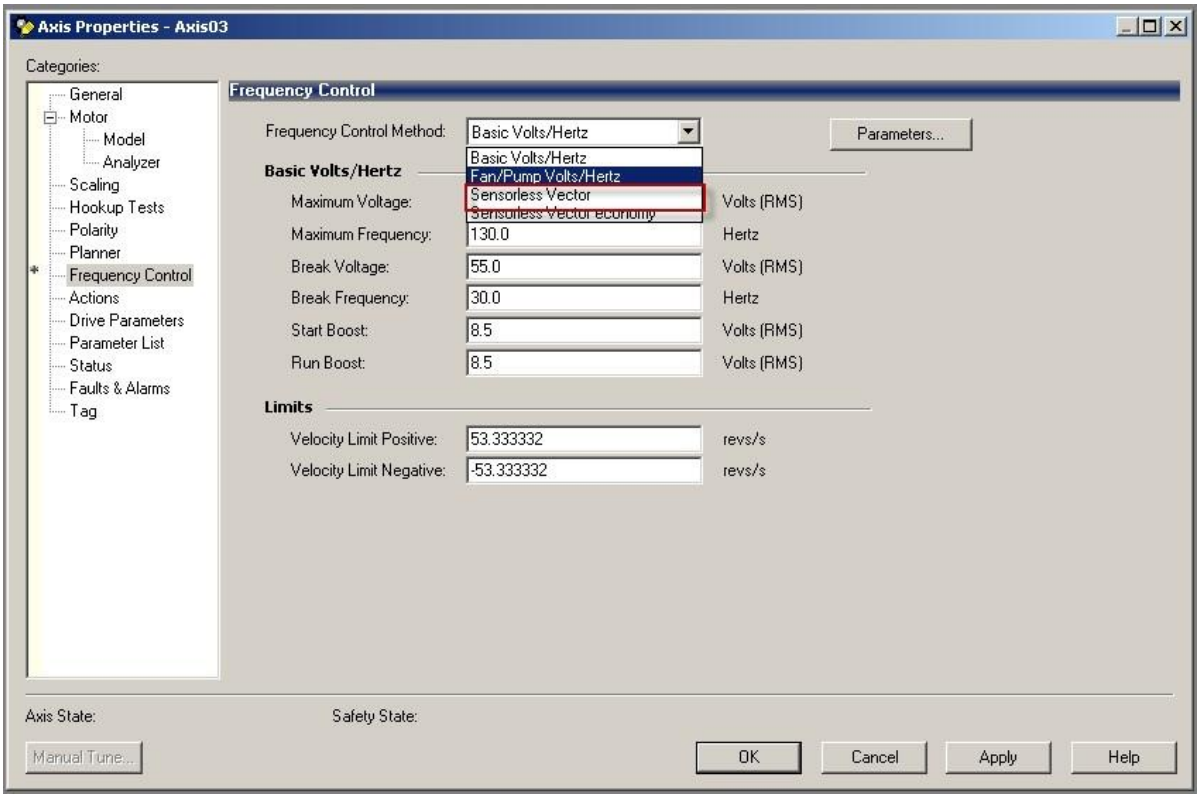
(2) Click **Apply** to save your changes.

Note: Position feedback will unwind or "rollover" once per motor revolution.

6. If a popup window appears, click **Yes** to automatically update all dependent attributes…

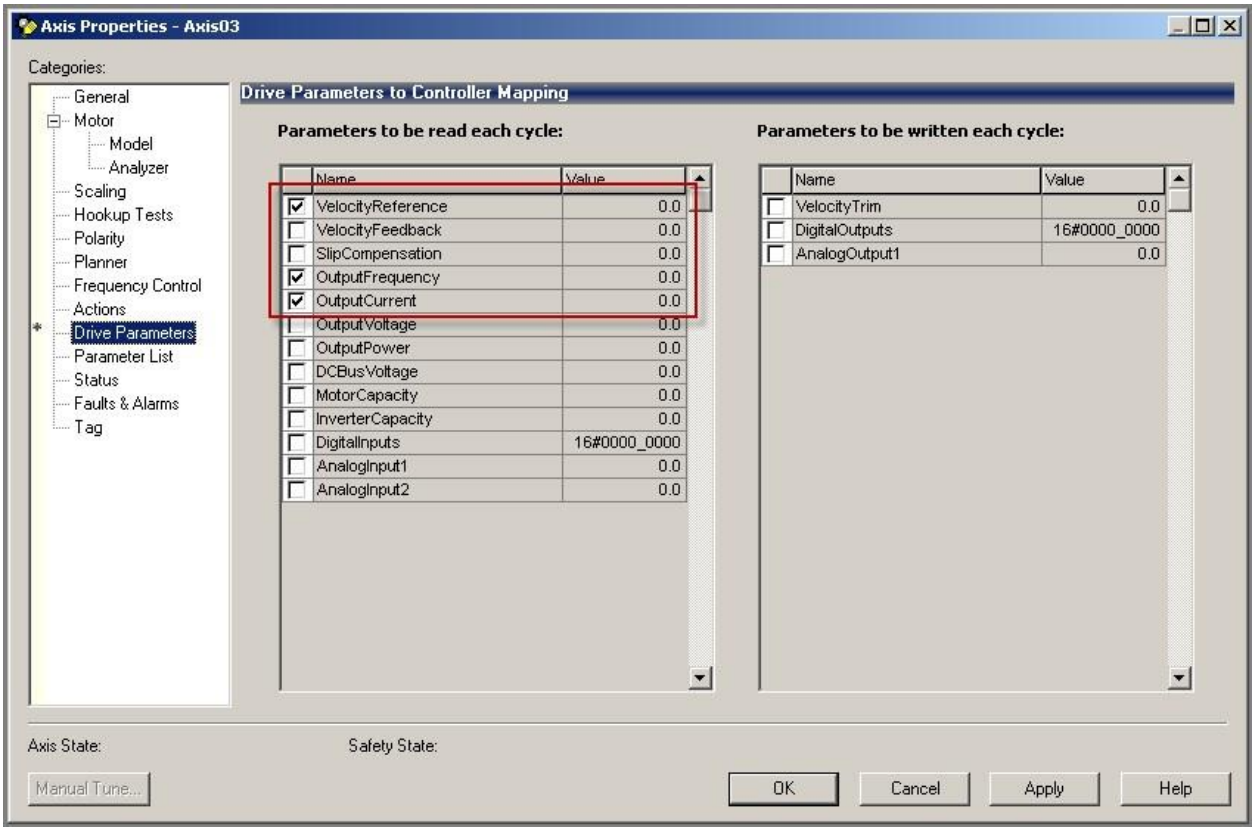

Logix Designer
You have changed a parameter which affects other attributes.
Do you want to automatically update all dependent attributes?
Refer to Help for a list of dependent attributes.
Yes
No
Help

7. Navigate to the **Frequency Control** page and select **Sensorless Vector** and click **Apply.**



Control Method	Description
Basic Volts/Hertz	Volts/Hertz control is a basic control method, providing a variable frequency drive for applications like fan and pump. It provides fair speed and starting torque, at a reasonable cost.
Fan/Pump Volts/Hertz	Fan/Pump Volts/Hertz is based on the Basic Volts/Hertz, but is specifically tailored for fan/pump applications.
Sensorless Vector	Sensorless Vector is an alternative Velocity Control Method that does not require configuration of a Volts/Hertz curve. Instead, by knowing the Stator Resistance and Leakage Inductance of the motor, the drive device can calculate the appropriate Output Voltage required for a given Output Frequency. This method provides better low speed Velocity Control behavior than by using the Basic Volts/Hertz method.
Sensorless Vector Economy	Induction Economizer mode consists of the sensorless vector control with an additional energy savings function. When steady state speed is achieved, the economizer becomes active and automatically adjusts the drive output voltage based on applied load. By matching output voltage to applied load, the motor efficiency is optimized. Reduced load commands a reduction in motor flux current.

8. Navigate to the **Drive Parameters** page ...



Notice you can select **Drive Parameters** in addition to the parameters included when **Auto Tag Update** is enabled.

Selected parameters can now be both “read” and “written” every coarse update rate.

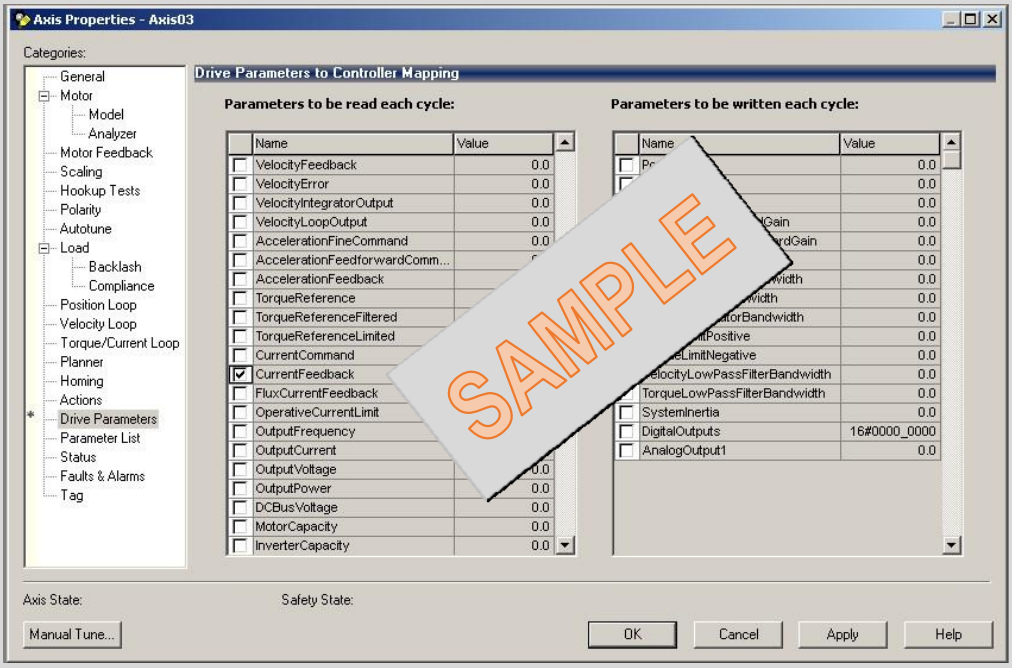
Scroll through the **read** parameter list and check ***‘VelocityReference’, ‘OutputFrequency’, ‘OutputCurrent’*** and then click

Apply to save your changes.

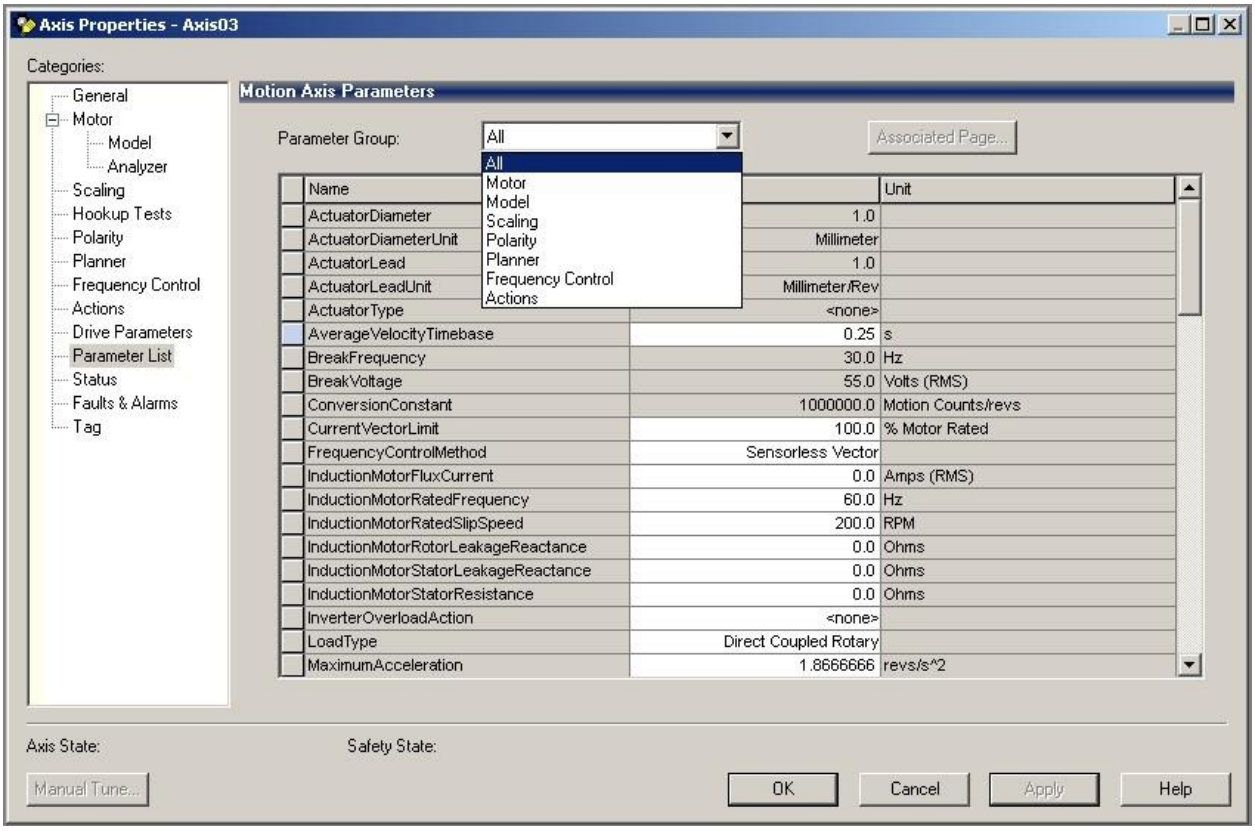
TIP:

Currently, there is a limit of 10-read and 10-write enabled selections per axis.

Each parameter selected to be transmitted as a cyclic read/write attribute will add overhead to the controller and drive data exchange and thus impact performance. You must analyze the trade-off of real time drive parameter exchange on the timing of the axes. The available drive parameters also depend on the motor control method that the axis is configured for. So while only a few show for Frequency Control, Velocity of Position Control exposes many more. Ex:



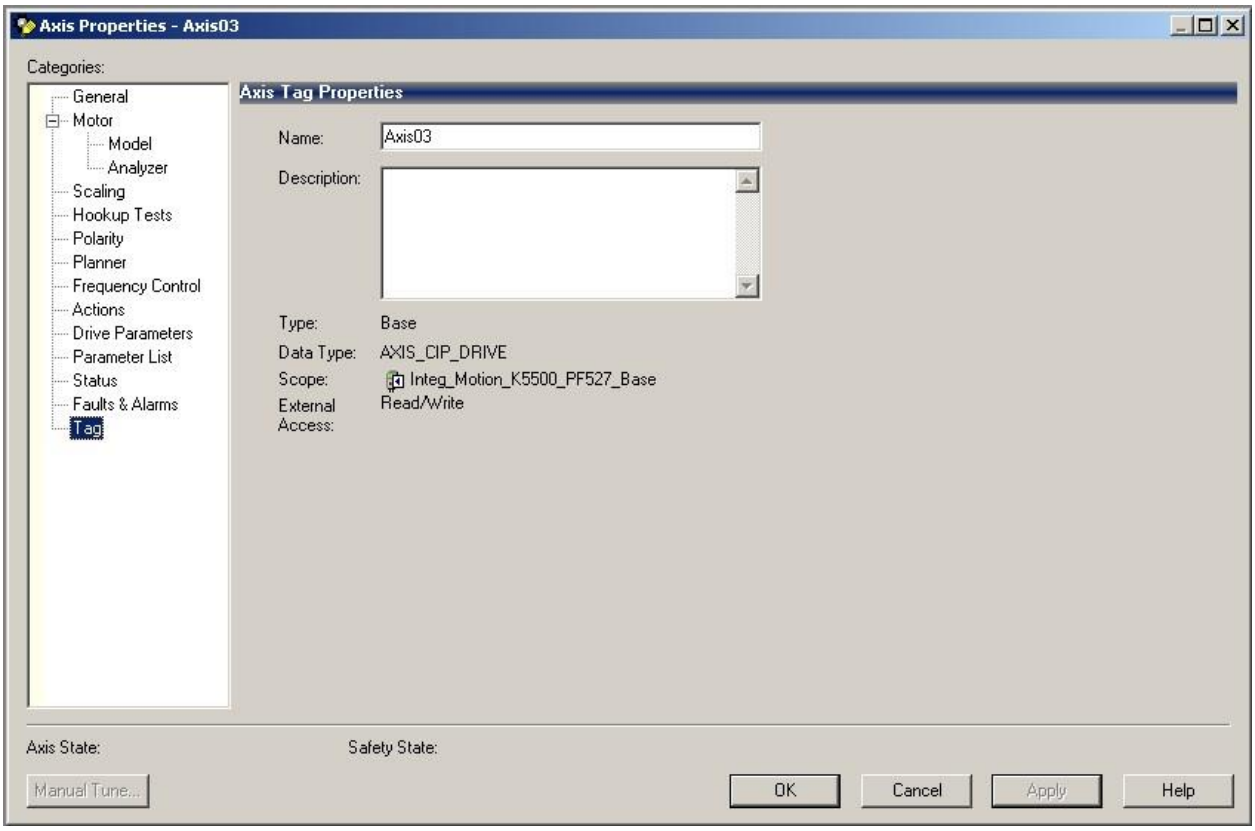
9. Navigate to the **Parameter List** page…



Notice you can access all the parameters associated with each category page. Take time to scroll through the various parameters.

> Each Parameter Group list may contain more attributes than the associated category page. In some instances, attributes listed on the Parameter Group list are not displayed on the associated category page. Also the parameters shown are dependent on the motor control configuration of the axis.

10. Navigate to the **Tag** page…



Notice that **Data Type** for an Integrated Motion on EtherNet/IP based drive is **AXIS_CIP_DRIVE**. This new data type was added in RSLogix 5000 v18 to support CIP Motion based drives.

11. Click **OK** to close the Axis Properties window.

Save and Download Your Motion Project

After completing the Logix configuration you must download your project to the CompactLogix controller.

1. Click the **Verify Controller** button on the Logix Designer toolbar.

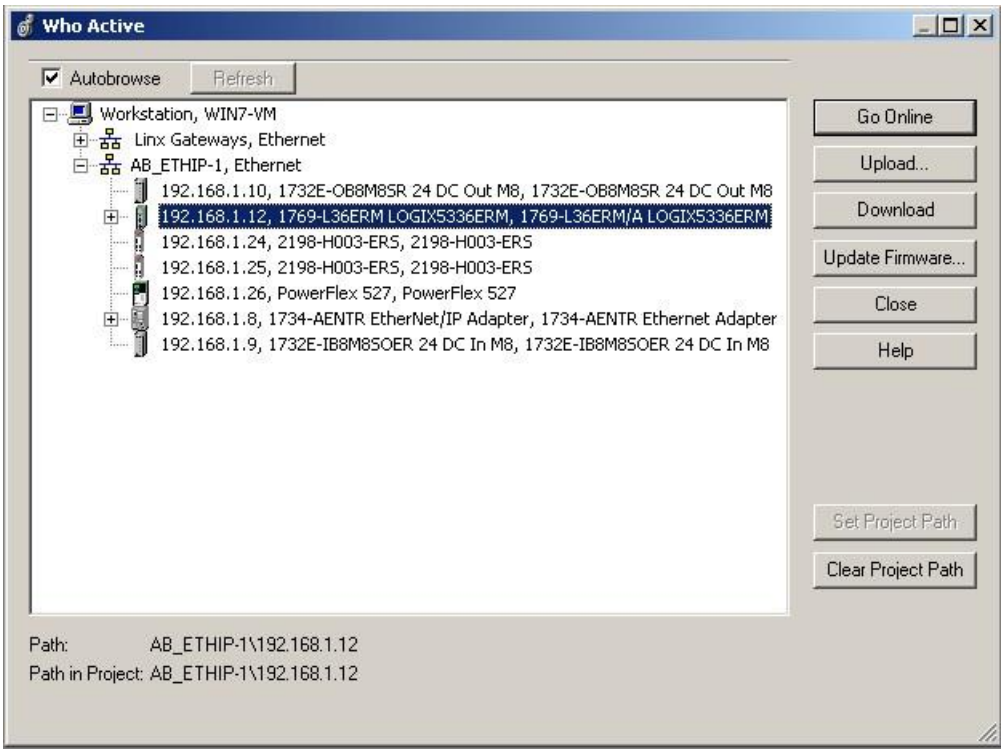
   The system verifies your Logix controller program and displays error/warnings, if any, in the status window.

2. Select **Save As…** from the **File** menu and save your program using a name of your choosing.

3. Select **Who Active** from the **Communications** menu.

   The Who Active window opens up.

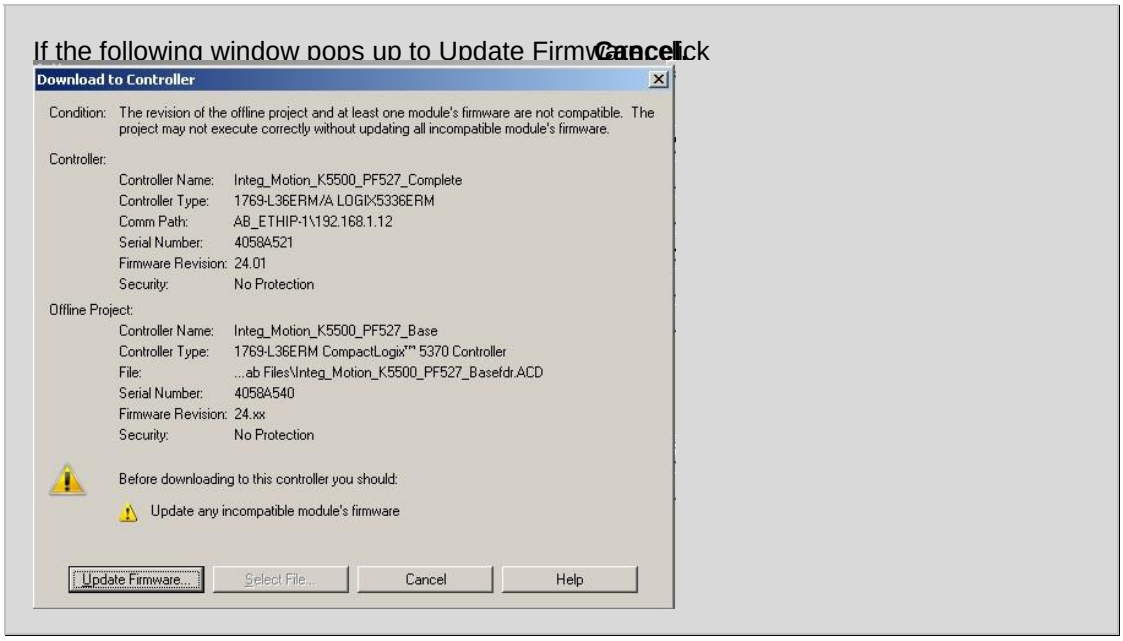
4. Drill down through the **AB_ETHIP-1** driver to find the processor at **192.168.1.12**…



Click **Set Project Path**.

5. Verify that the operation mode switch on your controller is in the **REM** (remote) position. Ensure that the **Kinetix 5500/PF527** demo is fully **"powered-up".**

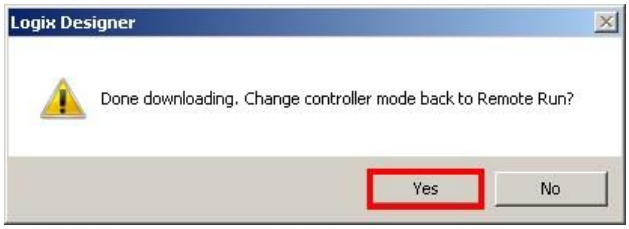
6. Click the **Download** button in the Who Active window.
   The Download window opens.


If the following window pops up to Update Firmware click Cancel
Download to Controller
Condition: The revision of the offline project and at least one module's firmware are not compatible. The project may not execute correctly without updating all incompatible module's firmware.
Controller:
Controller Name: Integ_Motion_K5500_PF527_Complete
Controller Type: 1769-L36ERM/A LOGIX5336ERM
Comm Path: AB_ETHIP-1\192.168.1.12
Serial Number: 4058A521
Firmware Revision: 24.01
Security: No Protection
Offline Project:
Controller Name: Integ_Motion_K5500_PF527_Base
Controller Type: 1769-L36ERM CompactLogix™ 5370 Controller
File: ...ab Files\Integ_Motion_K5500_PF527_Basefdr.ACD
Serial Number: 4058A540
Firmware Revision: 24.xx
Security: No Protection
Before downloading to this controller you should:
Update any incompatible module's firmware
Update Firmware...
Select File...
Cancel
Help

7. Click the **Download** button to send the program to the controller.

8. When downloading completes, place the controller back into **Run Mode**.

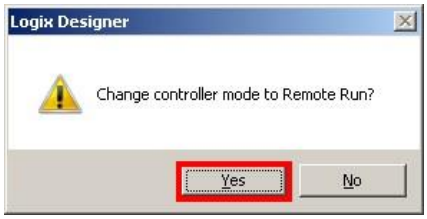
   Method 1:



   Click **Yes**.

   Method 2:

   Go to and select **Run Mode**



   Click **Yes**.

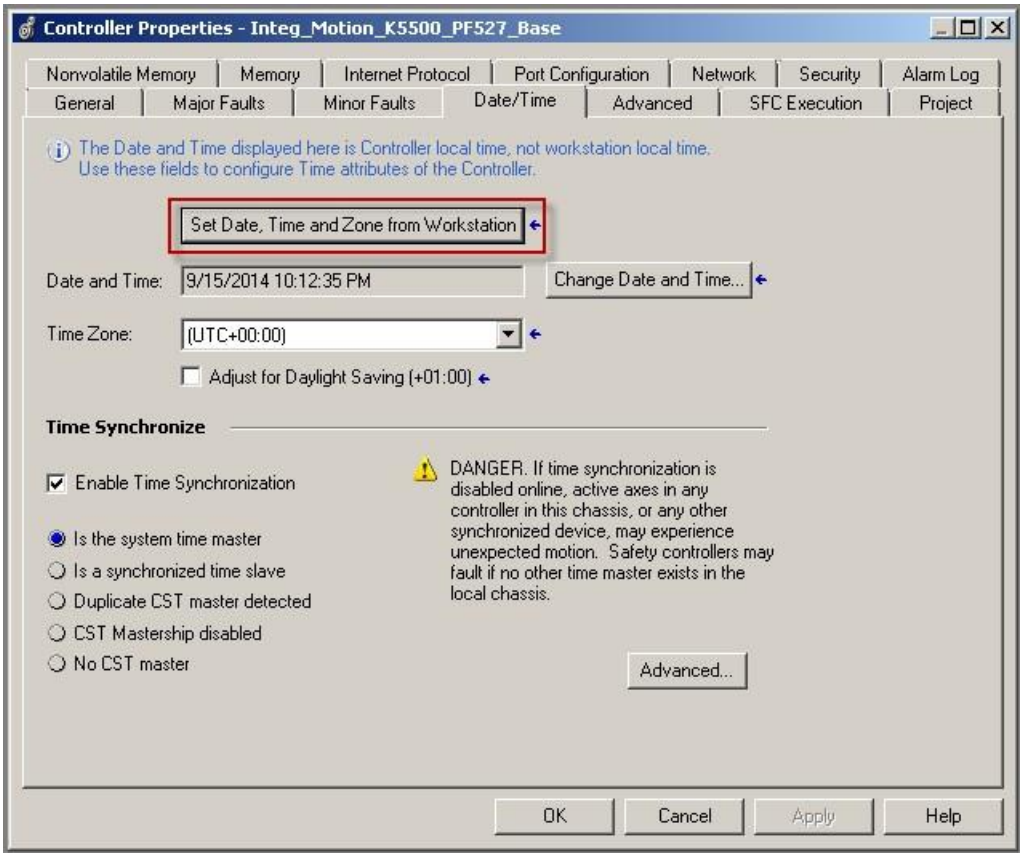
9. Verify that your controller is **"communicating"**...
   - **OK** indicator should be **solid green**.
   - **LINK1** indicator should be **flashing green**, indicating network activity
   - **LINK2** indicator should be **flashing green**, indicating network activity

10. Verify that the K5500 drives are **"ready"**…
    - **Module** light should be **solid green**.
    - **Network** light should be **solid green**.
    - **Port** status indicators may be **flashing green**, indicating network traffic.
    - The Kinetix 5500 and PF527 drives should both display **"STOPPED"** in all capital letters at the top of their displays.

    If any of the above steps did not work as described, please consult your lab instructor.

> Note: With the PF527 Beta units, the LEDs are not behaving properly. This issue has already been fixed. If your drive says STOPPED you are successfully connected.

11. Open the **Controller Properties (Edit Menu)** and navigate to the **Date/Time** tab…



Click the **'Set Date, Time and Zone from Workstation'** to set the current date and time.
Click **OK**.

12. **Save** your project again, and continue on to Lab 2.

You may be prompted to upload the tags from the controller – either selection will work.

Lab 2: Axis Commissioning – Hookup Test and Autotune (Estimated Time 10 Minutes)

This lab is built on the project file from Lab 1. In this lab we will introduce you to the process of commissioning a servo axis by performing the following:

- Axis properties Hookup Tests
- Axis properties Autotune
- Review common commissioning errors that may occur during an Autotune

Axis Hookup Tests

In this section of the lab, you will use Logix Designer to access the Axis Properties to run the Motor and Feedback Hookup Test. The Motor and Feedback test applies motion to the motor allowing the user to verify the power and feedback connections between the drive and motor. This test also establishes the forward or positive direction of axis motion.

1. You should be **Online** with your controller.

2. Before running the Motor and Feedback Hookup Test verify that the K5500 drives are **"ready"**.

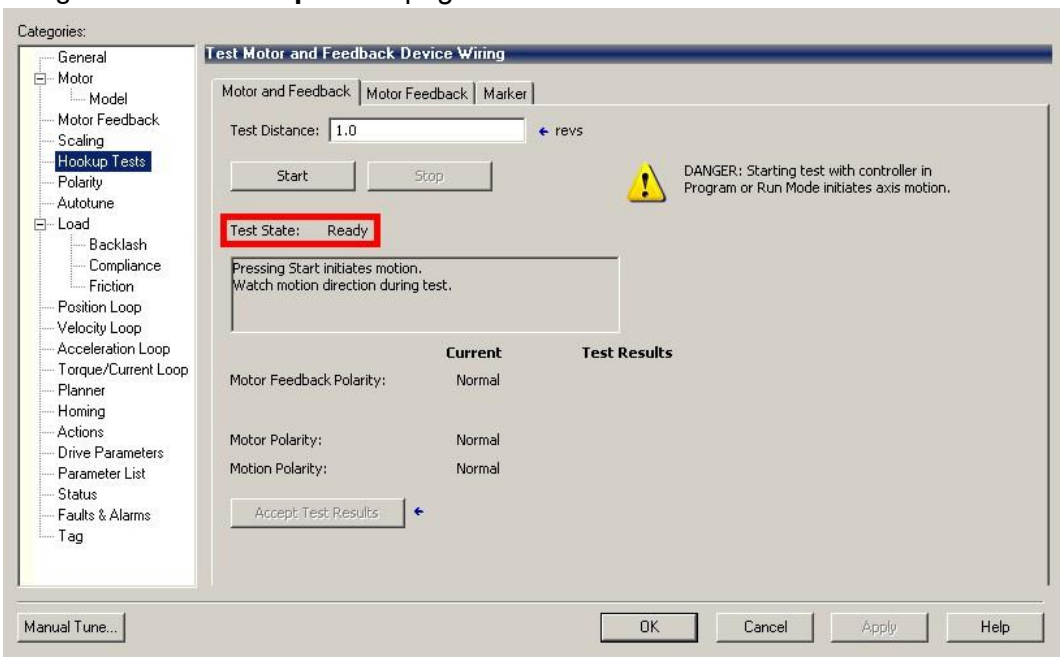
   Check the drive status:

   - **Module** light should be **solid green**.
   - **Network** light should be **solid green**.
   - **Port** status indicators may be **flashing green**, indicating network traffic.
   - The Kinetix 5500 and PF527 drives should both display **"STOPPED"** in all capital letters at the top of their displays.

3. Right-click on **Axis02** and select **Properties**…

   The Axis Properties window opens.

4. Navigate to the **Hookup Tests** page…



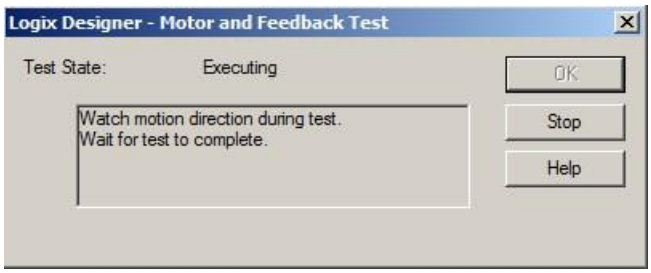
   Enter a **Test Distance** of ***'1.0'*** revs. This will provide enough axis travel to detect a marker. **Test State** should display **Ready**.

   The Hookup Tests make the axis move even when the controller is in program mode.

5. Press **Start** to conduct the test.

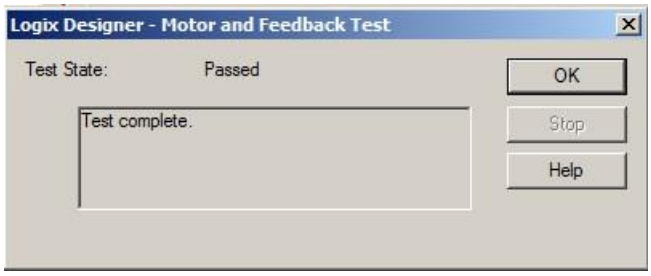
   Once the Start button has been pressed, the axis will immediately begin to move.

6. The Motor and Feedback Test window opens…



You will hear the servo enable and you should observe Axis02 move approximately one revolution in the CW direction.

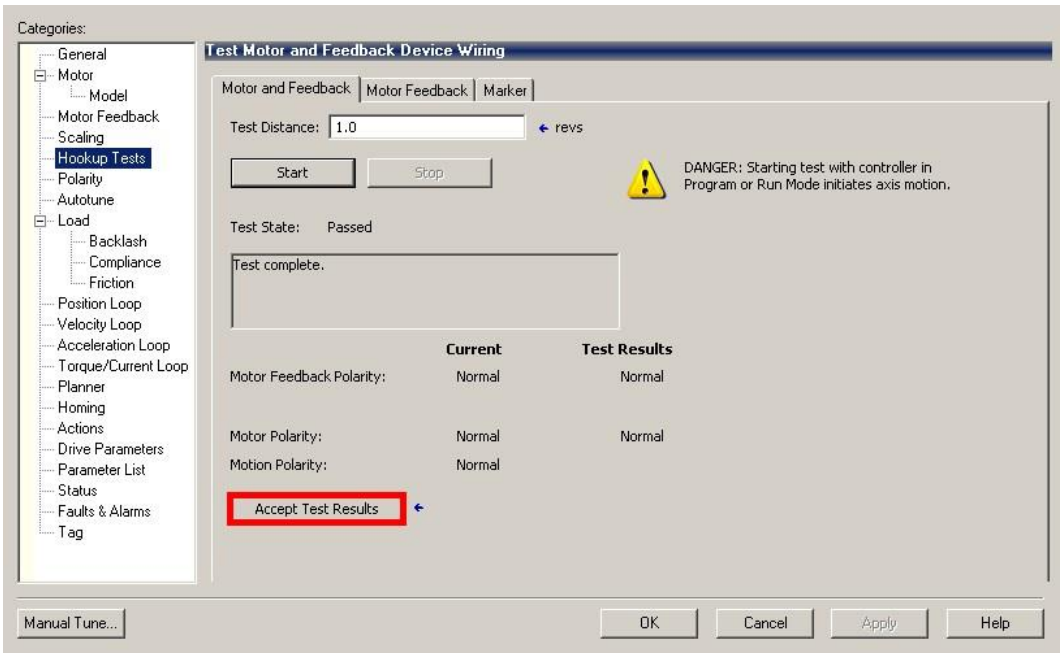
7. When the motor has completed one revolution and the drive has received the encoder signals correctly, the **Test State** will change from **Executing** to **Passed**…



Click **OK**.

8. Click **Yes** if the axis moved in the forward or positive (CW) direction…

9. Click **Accept Test Results** to update/save the Motor and Feedback Polarities…



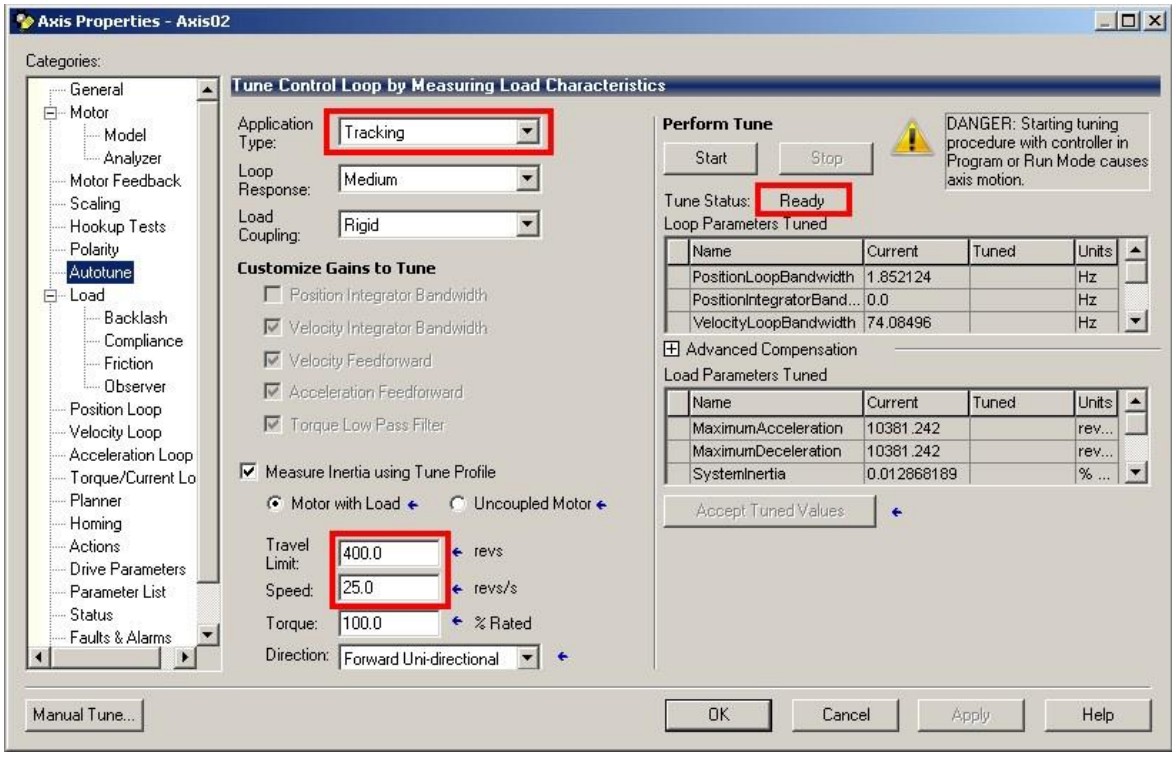
10. Proceed to the Axis Autotune section of the lab.

Axis Autotune

In this section of the lab, we will tune Axis02. The Autotune measures the system inertia, acceleration/deceleration rates, as well as calculates the position/velocity regulator gains.

1. If the Properties window is not open, right-click on **Axis02** and select **Properties**.
   The Axis Properties window opens.

2. Navigate to the **Autotune** page…

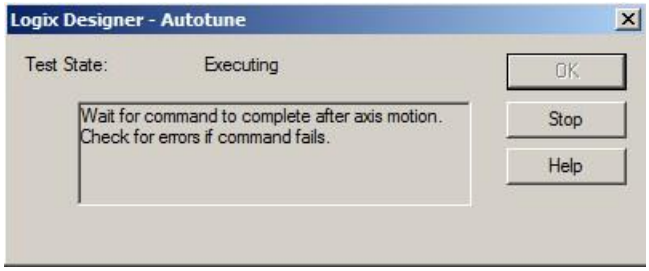


Set the **Application Type** to ***'Tracking'***, the **Travel Limit** to ***'400'*** revs and the **Speed** to ***'25'*** revs/s. **Tune Status** should display **Ready**.

3. Click **Apply** and press **start** to initiate the Autotune.

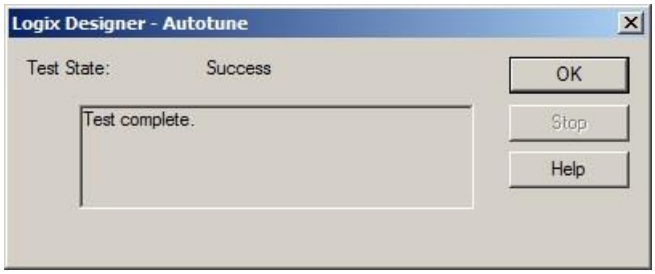
   Once the Start button has been pressed, the axis will immediately begin to move.

4. The Autotune window opens…



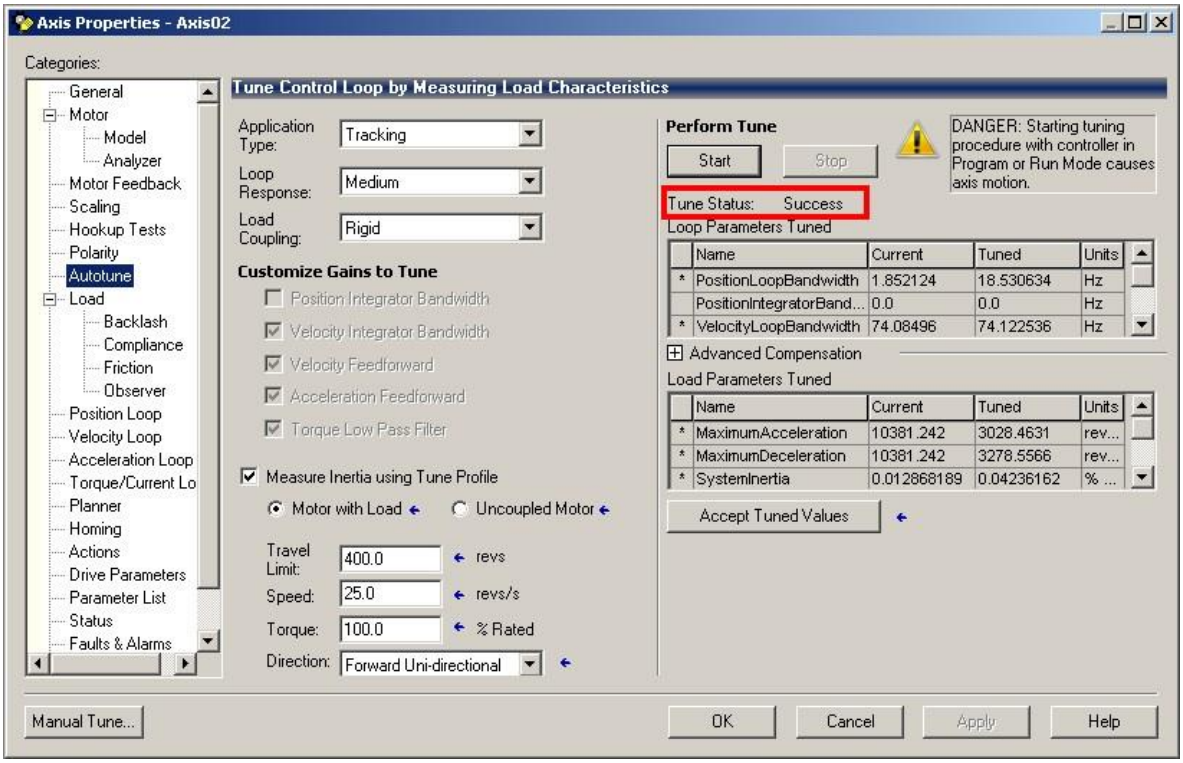
You should hear the servo enable for as long as it takes to reach the configured speed and then decelerate. This is a very quick process, usually less than one second.

5. When the Autotune completes, the Test State will change from **Executing** to **Success**…



Click **OK**.

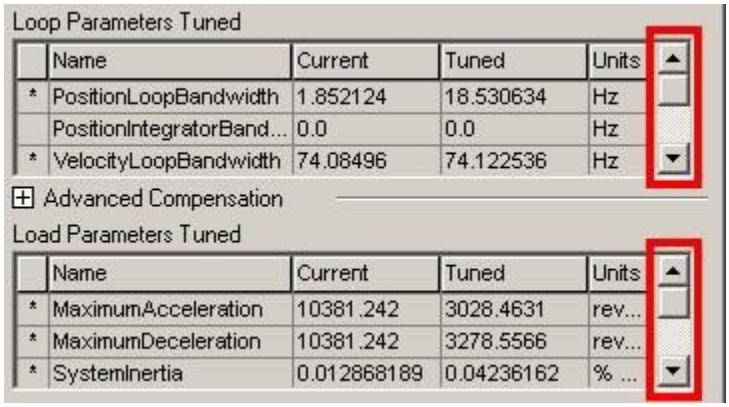
6. Your **Tune Status** should display **Success**…



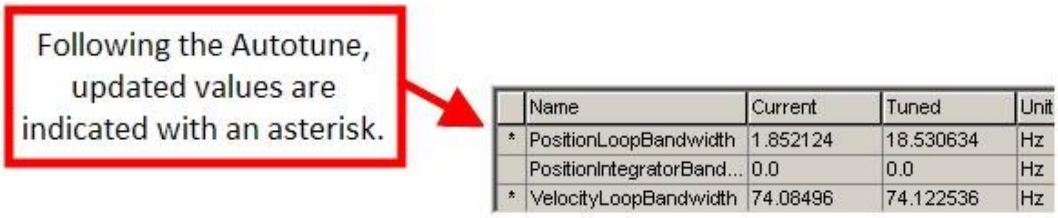
If your Tune Status does not display **Success**, please refer to the Common Faults Encountered While Tuning section in a few pages.

If you have any questions, please consult with your lab instructor.

7. Take time to scroll through the **Loop Parameters Tuned** and **Load Parameter Tuned** lists…

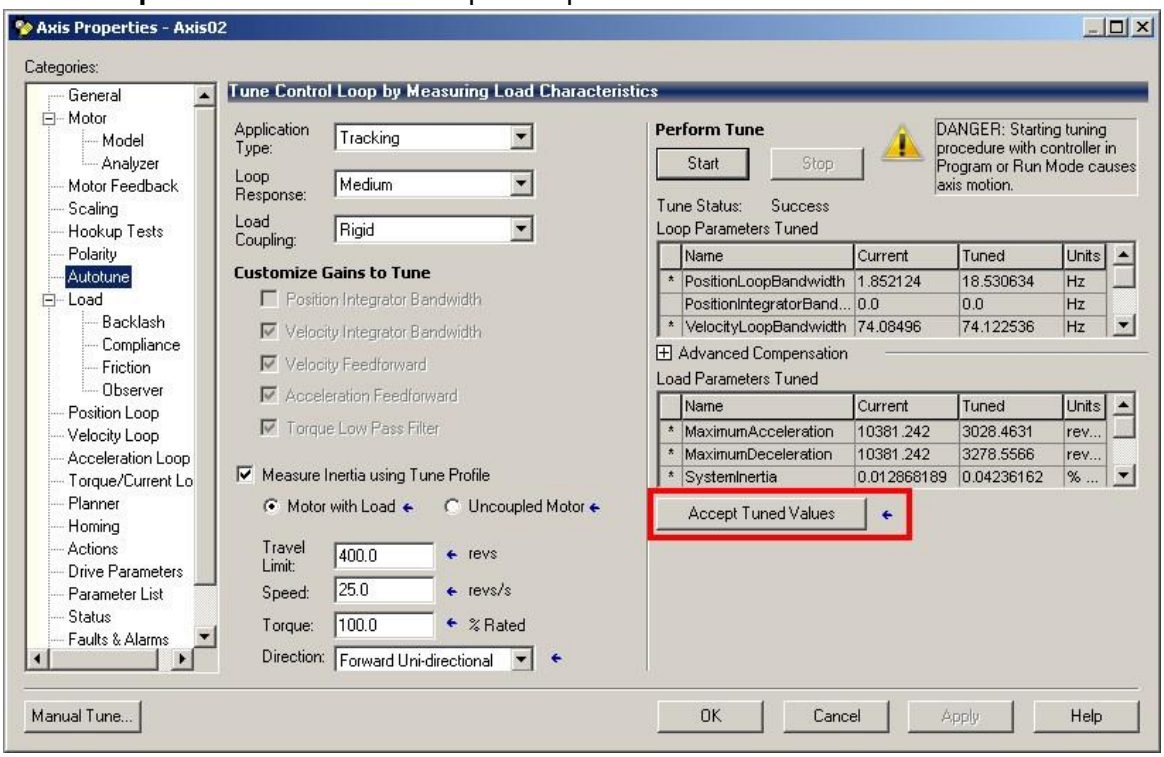

Loop Parameters Tuned
Name | Current | Tuned | Units
* PositionLoopBandwidth | 1.852124 | 18.530634 | Hz
PositionIntegratorBand... | 0.0 | 0.0 | Hz
* VelocityLoopBandwidth | 74.08496 | 74.122536 | Hz
Advanced Compensation
Load Parameters Tuned
Name | Current | Tuned | Units
* MaximumAcceleration | 10381.242 | 3028.4631 | rev...
* MaximumDeceleration | 10381.242 | 3278.5566 | rev...
* SystemInertia | 0.012868189 | 0.04236162 | % ...

Notice which parameters were updated following the Autotune.


Following the Autotune, updated values are indicated with an asterisk.
Name | Current | Tuned | Unit
* PositionLoopBandwidth | 1.852124 | 18.530634 | Hz
PositionIntegratorBand... | 0.0 | 0.0 | Hz
* VelocityLoopBandwidth | 74.08496 | 74.122536 | Hz

Note: The Current and Tuned values are both displayed, indicating the "before and after" Autotune results.

8. Click **Accept Tuned Values** to accept the updated Autotune values…



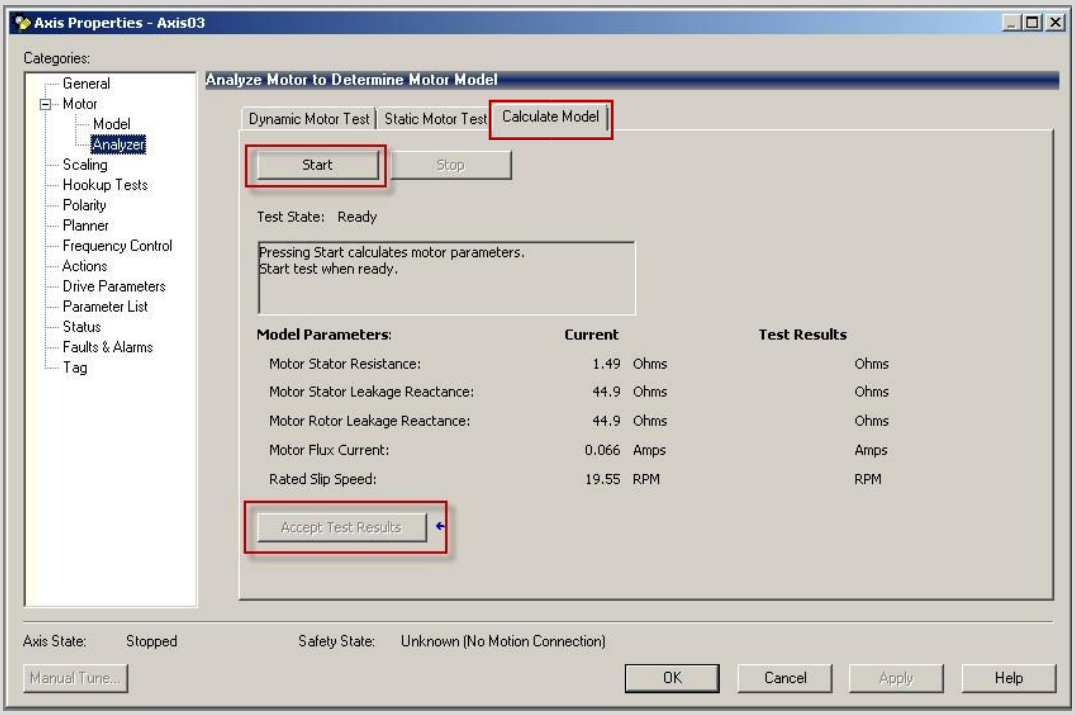
9. Click **OK** to close the Axis Properties window.

10. **Save** your project.

    You may be prompted to upload the tags from the controller – either selection will work.

11. ***IMPORTANT! You MUST Perform this step for the PF527*** Using what you learned in this section, execute a **Calculate Tune** on the PowerFlex 527 **Axis03**

TIP: You'll need to navigate to the **Motor->Analyzer->Calculate Model** section of the Axis03



Use the Motor Analyzer tool to identify the model for motors that have the data source set to Nameplate Datasheet. For all other motor data source configurations, this test is not applicable. The Motor Analyzer dialog box applies for PowerFlex drives and supports Induction and Permanent Motor types.

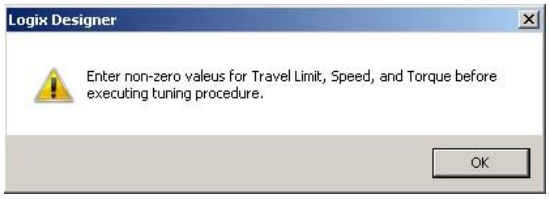
The Motor Analyzer dialog box contains a number of tests that can be executed - each contained within separate tabs on the Motor Analyzer dialog box. Each of the tests is similar in that they each consist of a Start, a Stop, an Information, and an Accept Test Result control. Test Results display an output of test execution.

12. **Save** your project.

Common Faults Encountered While Tuning

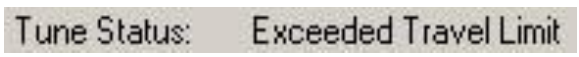
In this section we will review some of the more common faults that are encountered during an Autotune.

1. Autotune Travel Limit, Speed, or Torque set to zero…



   - Check to make sure that the Autotune Travel Limit, Speed, and Torque are all set to a non-zero value.

2. Exceeded Travel Limit…



- The Autotune Speed might be set too high, check the speed and decrease it.
- The Autotune Travel Limit might be set too low, check the test distance and increase it.

Lab 3: Using Motion Direct Commands (Estimated Time 10 Minutes)

Motion Direct Commands let you issue motion commands without having to write or execute an application program. You must be online with your controller to execute a Motion Direct Command. Let's see how these work using the project you created in the previous labs.

Jogging an Axis Using Motion Direct Commands

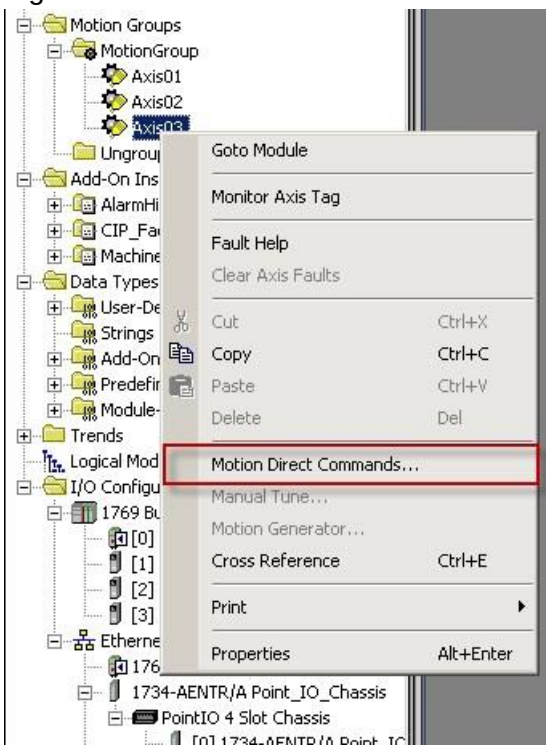
1. Before running the Motion Direct Commands, verify the drives are **"ready"**.

   Check the K5500 drive status:

   - **Module** light should be **solid green**.
   - **Network** light should be **solid green**.
   - **Port** status indicators may be **flashing green**, indicating network traffic.
   - The Kinetix 5500 and PF527 drives should both display **"STOPPED"** in all capital letters at the top of their displays.

   Note: With the PF527 Beta units, the LEDs are not behaving properly. This issue has already been fixed. If your drive says STOPPED you are successfully connected.

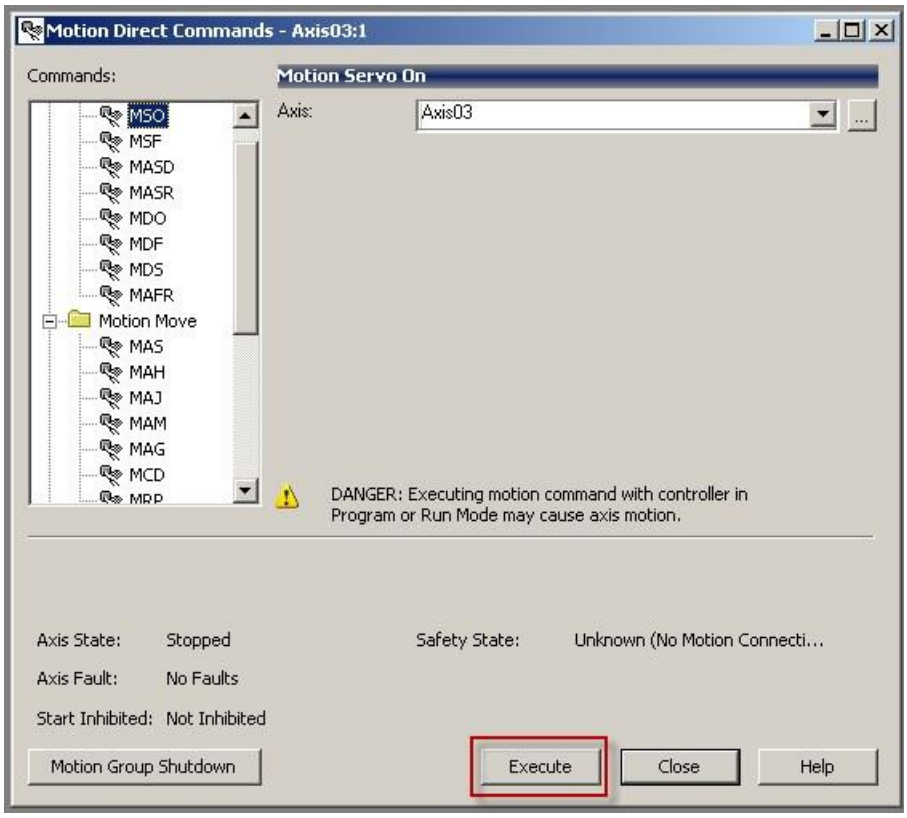
2. Right-click on **Axis03** and select **Motion Direct Commands**…



   The Motion Direct Commands window opens.

3. Take a moment to look through all the commands available to you by moving the mouse cursor over the instructions.

4. Select the **Motion Servo On (MSO)** instruction…

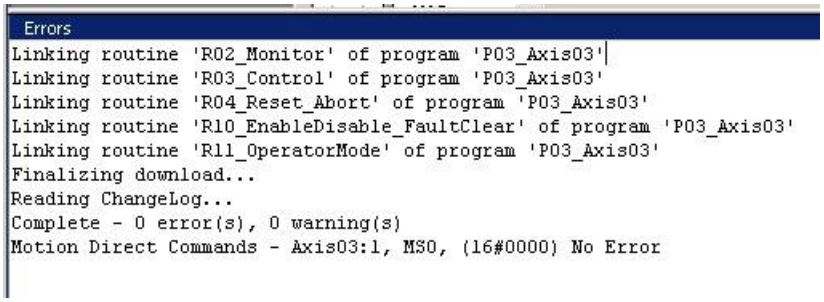


The MSO instruction enables the specified axis by activating both the drive amplifier and drive control loop.

****IMPORTANT** If the motor spins after only enabling the drive, go back to Step 11 in the previous lab section to execute a Calculate tune.**

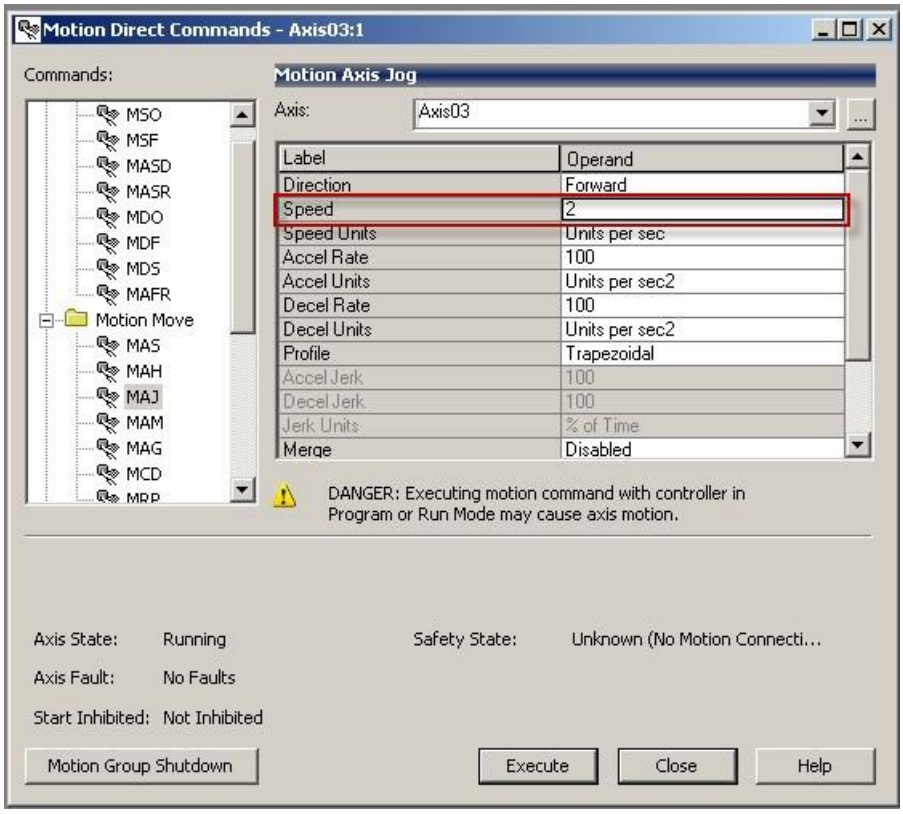
5. Click **Execute**.

6. You should see an indication that the command was executed in the Errors window and hear the PF527 drive fan turn on…


Errors
Linking routine 'R02_Monitor' of program 'P03_Axis03'
Linking routine 'R03_Control' of program 'P03_Axis03'
Linking routine 'R04_Reset_Abort' of program 'P03_Axis03'
Linking routine 'R10_EnableDisable_FaultClear' of program 'P03_Axis03'
Linking routine 'R11_OperatorMode' of program 'P03_Axis03'
Finalizing download...
Reading ChangeLog...
Complete - 0 error(s), 0 warning(s)
Motion Direct Commands - Axis03:1, MSO, (16#0000) No Error

You should also notice that the display shows the drive status as “Running”.

7. Select the **Motion Axis Jog (MAJ)** instruction…



Enter in a **Speed** value of **'2'**.

The MAJ instruction will move an axis at a constant speed until you tell it to stop.

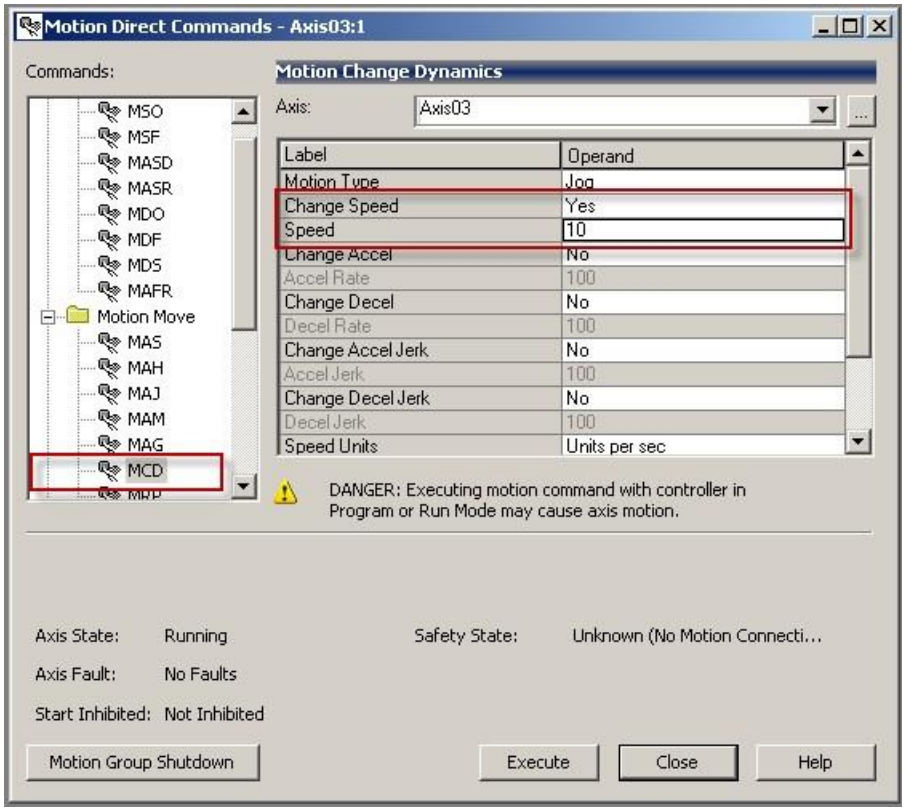
8. Click **Execute**.

   Once the Execute button has been pressed, the axis will immediately begin to move.

9. The axis should be rotating at **'2 revs/s'**. Though this speed can be monitored in the controller, you can verify this visually.

Varying the Speed of the Axis Using a Motion Direct Command

1. Select the Motion Change Dynamics (MCD) instruction…



Set **Change Speed** to **Yes** and enter a **Speed** of ***'10'***.

The MCD instruction will selectively change the speed, acceleration rate, or deceleration rate of a move and/or jog profile in process.

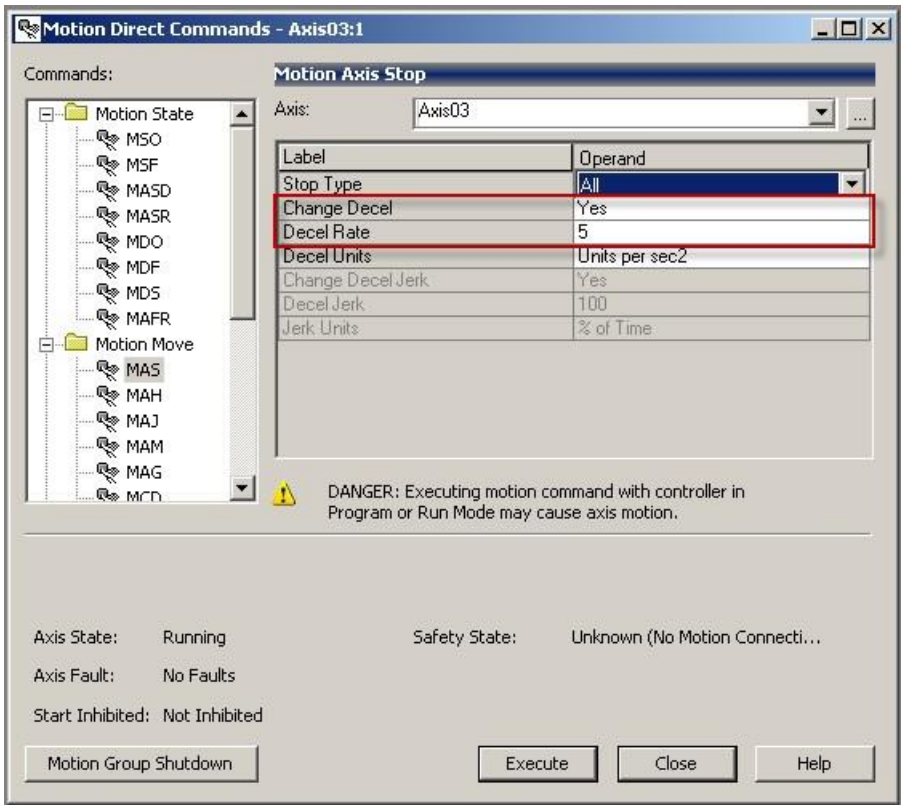
2. Click **Execute**.

   You should see a clear increase in the rotational speed of the axis.

   Remember, we initially had configured the axis to jog at 2 revs/s. Now it's rotating at five times that speed and without having to write an application program – everything was done "on the fly" using Motion Direct Commands!

Stop the Axis Using a Motion Direct Command

1. Select the **Motion Axis Stop (MAS)** instruction…



Set **Change Decel** to **Yes** and enter in a **Decel Rate** of ***'5'*** Units per sec2, or revs/s^2.

The MAS instruction will initiate a controlled stop of any motion process on the designated axis.

2. Click **Execute**.

3. When the axis has slowed to a stop, select and execute the **Motion Servo Off (MSF)** instruction.

   The MSF instruction disables the specified axis by deactivating both the drive amplifier and the drive control loop.

4. Click **Execute** the PF527 should now show a STOPPED status on the HIM.

5. Now use the **Motion Drive Start (MDS)** instruction. If you have any questions please consult with your lab instructor.

   Note: You will **NOT** have to execute an MSO instruction to enable the axis again. Execute an MAS to stop the axis and a MSF instruction to disable the axis when you are finished.

> **MDS (Motion Drive Start)** Instruction supports the Kinetix 6500/5500 drive in Torque Mode, or the PowerFlex drive in Torque Mode or Velocity Mode.
> Once either drive is put in Direct Control mode, the following motion instructions are not allowed: MSO, MRP, MAH, MAPC, MATC, MCT, MAG. When the drive is in Direct Torque Mode, the drive is controlled with a TorqueOffset. When the drive is in Direct Velocity Mode, the drive is controlled by the RampRate, and other velocity attributes associated only with the PowerFlex drive.

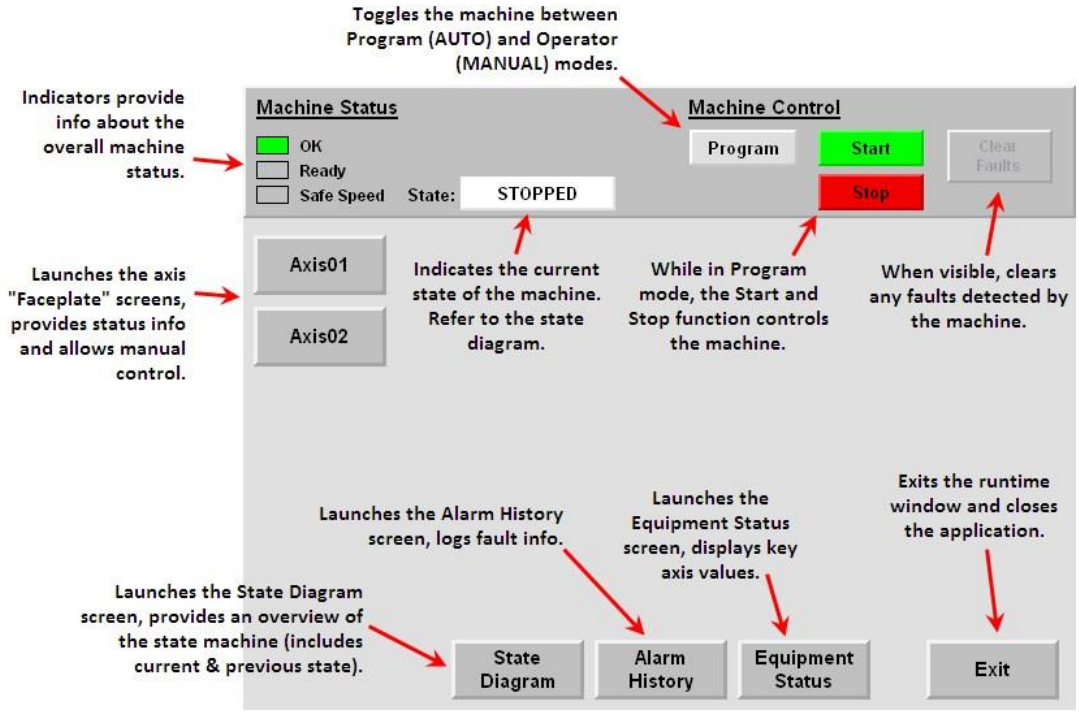
Lab 4: Adding an HMI (Estimated Time 15 Minutes)

The following lab previews the *'Drives and Motion Accelerator Toolkit'* FactoryTalk View ME file to control your motion application. There are several preconfigured HMI screens that were designed specifically for drives and motion applications, including:

- Standard preconfigured auto/manual control templates
- Predefined axis status templates
- Preconfigured fault/diagnostic templates

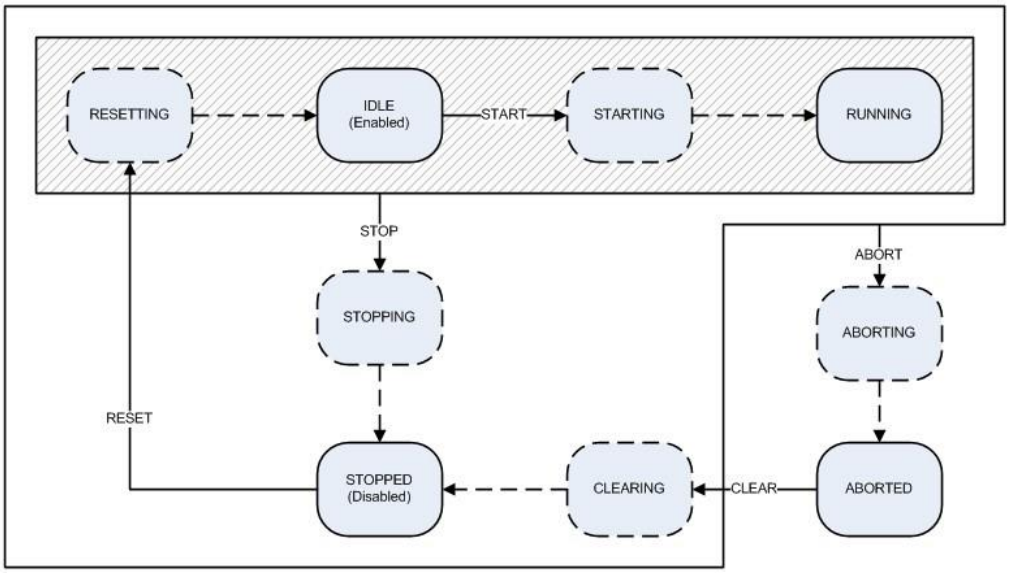
Switch to the HMI Application

1. **Minimize** Logix Designer so that the HMI screen on the desktop can be seen.
2. If the warning screen is displayed, click **Start The Lab** to load the **Startup** screen.
3. The **Startup** screen should initially be displayed…


Toggles the machine between Program (AUTO) and Operator (MANUAL) modes.
Indicators provide info about the overall machine status.
Machine Status
OK
Ready
Safe Speed
State:
STOPPED
Machine Control
Program
Start
Stop
Clear Faults
Launches the axis "Faceplate" screens, provides status info and allows manual control.
Axis01
Axis02
Indicates the current state of the machine. Refer to the state diagram.
While in Program mode, the Start and Stop function controls the machine.
When visible, clears any faults detected by the machine.
Launches the Alarm History screen, logs fault info.
Launches the Equipment Status screen, displays key axis values.
Exits the runtime window and closes the application.
Launches the State Diagram screen, provides an overview of the state machine (includes current & previous state).
State Diagram
Alarm History
Equipment Status
Exit

The **Startup** screen provides Machine Status and Control, plus it allows navigation to all other screens. Take a moment to familiarize yourself with the Startup screen before moving on to the next section. It may look a little different, depending on the status of the machine.

4. While in **Program** mode, the Machine operates based on the following state diagram…



The states with a dashed outline indicate a transitional state, while those with a solid outline indicate an end state.

Depending on your current machine state, use the following commands to transition between states:

ABORTED – Press **Clear Faults**, *ABORTED -> CLEARING -> STOPPED*

STOPPED – Press **Start**, *STOPPED -> RESETTING -> IDLE -> STARTING -> RUNNING*

RUNNING – Press **Stop**, *RUNNING -> STOPPING -> STOPPED*

Note: The machine is placed into the ABORTED state whenever a drive fault condition and/or a state transition error has been detected. The machine is also placed into the ABORTED state on Power Up or during "first scan" (i.e. Program to Run Mode) of the controller. Refer to the **Alarm History** faceplate to determine the cause for the ABORTED condition.

Faceplate Operation

Machine Control

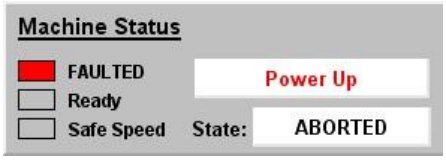
Program (AUTOMATIC) mode refers to the automatic function or automatic sequencing for the machine. Operator (MANUAL) mode allows for some manual operations, like enable, disable, move, jog, home, etc. The machine status indicators provide a summation view of all the devices for the entire machine. The Program/Operator selector button lets you toggle between the two modes.

The Clear Faults button attempts to clear faults on all devices. The condition that caused the fault must be corrected before the clear is successful.

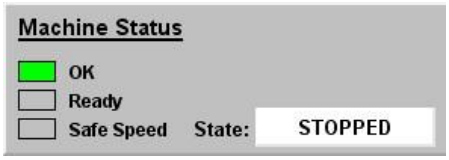
Run the Machine

Follow these steps to start and stop the motion system in Program mode.

1. If the machine is currently in the **ABORTED** state…



…press **Clear Faults**.



After a few moments the machine should transition to the **STOPPED** state.

2. Press **Program/Operator** button until **Program** is displayed. Program (AUTO) is now the active control mode.

3. Press **Start**.

   The required axes enable and begin operating according to the Logix Designer program.

4. Press **Stop**.

   The motion system stops.

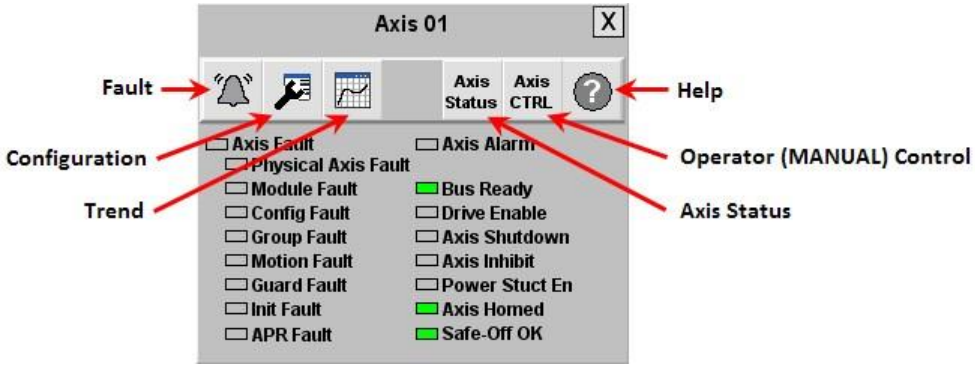
5. Press Program/Operator.

   The machine must be stopped before you can switch control modes. When in Operator (MANUAL) mode, you can individually control each axis from its corresponding faceplate.

CIP Motion Axis Faceplate

1. Press the

button from the **Startups** screen to launch the faceplate…

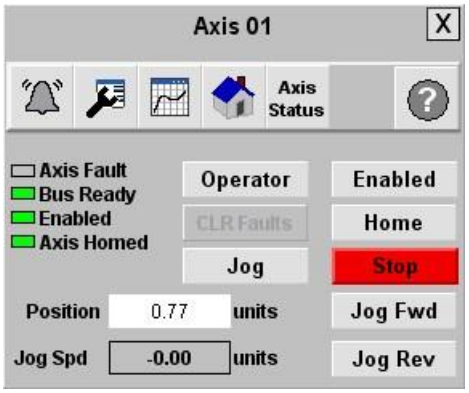


The CIP Motion Axis faceplate provides axis status information, fault information, and trending data. The faceplate also includes the ability to manually control the axis.

These screenshots were done with Axis01, however Axis02 that you just configured could be used as well.

2. Press

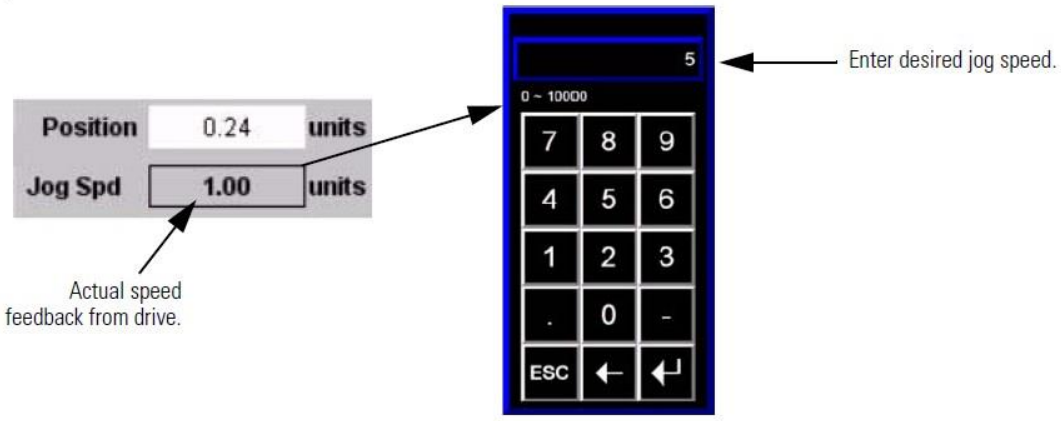
button…



From the **Axis CTRL** display, you have the ability to enable, disable, home, move, jog, and clear an axis fault.

Note: Some of the numbers shown are both indicators and keypad input buttons. For example, the Jog Spd indicator displays the actual speed feedback of the drive, not the desired jog speed. However, by clicking the indicator you launch the keypad input object where you can enter the desired jog speed. These inputs will only work when the axis is in Operator mode and the drive is Enabled.

Dual Purpose Controls

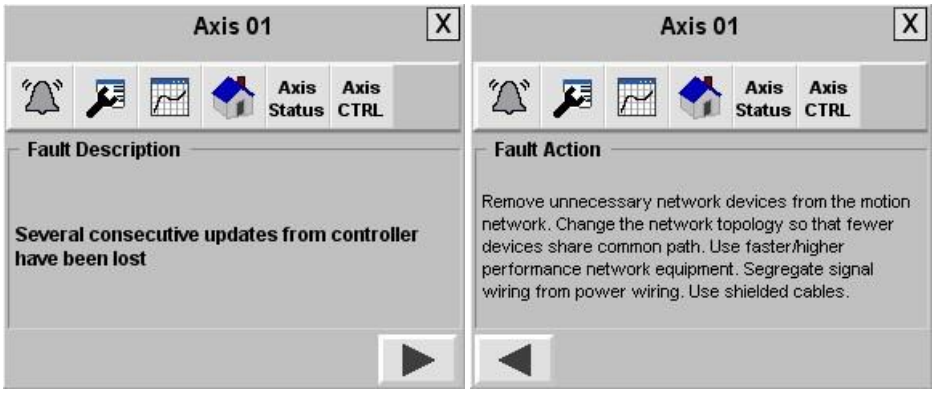


3. Press the button…

If a fault condition exists, the icon **flashes** yellow. The **Fault** display determines the fault information from the drive and displays the fault type, code, and description. If there is no active fault, the display shows the last fault condition recorded.

To easily generate the fault shown on the screenshot, remove the Ethernet cable from its port on one of the drives.

4. Press the button…


Axis 01
Axis Status
Axis CTRL
Fault Description
Several consecutive updates from controller have been lost
Axis 01
Axis Status
Axis CTRL
Fault Action
Remove unnecessary network devices from the motion network. Change the network topology so that fewer devices share common path. Use faster/higher performance network equipment. Segregate signal wiring from power wiring. Use shielded cables.

The **Help** screen displays the fault descriptions and actions. Press the arrows to switch between screens.

You can clear faults from the **Startup** screen or, if in Operator mode, from the **Axis CTRL** display. The Alarm History screen logs fault information from all of the devices.

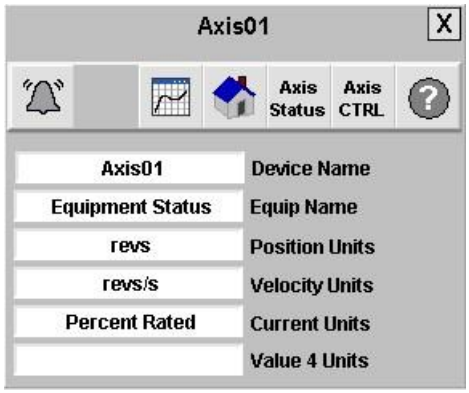
When you are finished, reconnect the Ethernet cable to the drive.

Note: The drive will automatically recover from a **'Control Sync Fault'**, but the machine is still faulted. Therefore, a Clear Faults command on the machine will be needed once the Ethernet cable has been reconnected. It may take up to a minute to recover.

5. Press Clear Faults…
6. Press **Program/Operator** until **Operator** is displayed and then again until **Program** is displayed.

   This is required because we had Operator control of the axis above while we were jogging it manually.

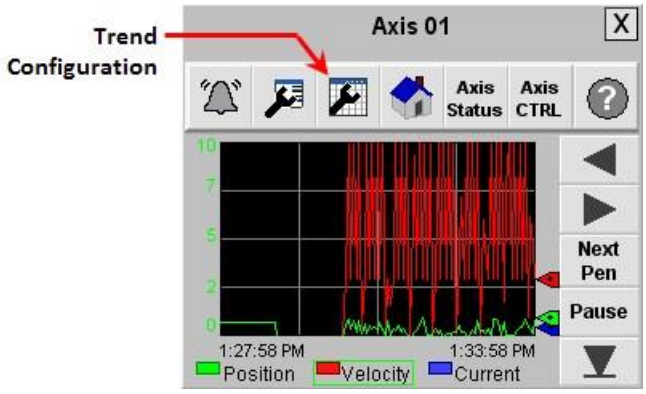
7. Press **Start**.
8. Press the button…



From the **Configuration** screen you can enter display names and units as required for your application.

Some of the labels are used on the **Equipment Status** faceplate.

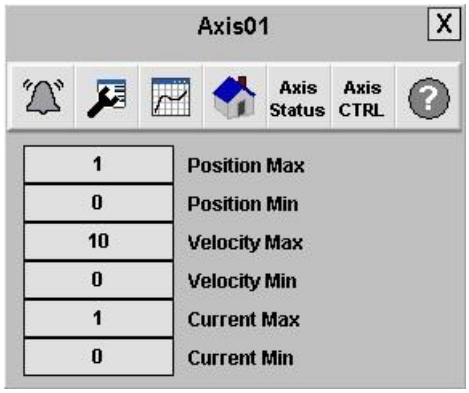
9. Press the button…



The **Trend** screen lets you view your current feedback, actual velocity, and actual position trends of your axis.

The **Trend Configuration** button is only visible on the **Trend** screen.

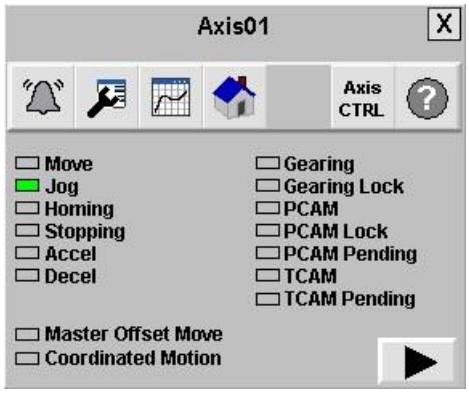
10. Press the button…



The **Trend Configuration** screen lets you adjust the trend scales.

11. Press

button…

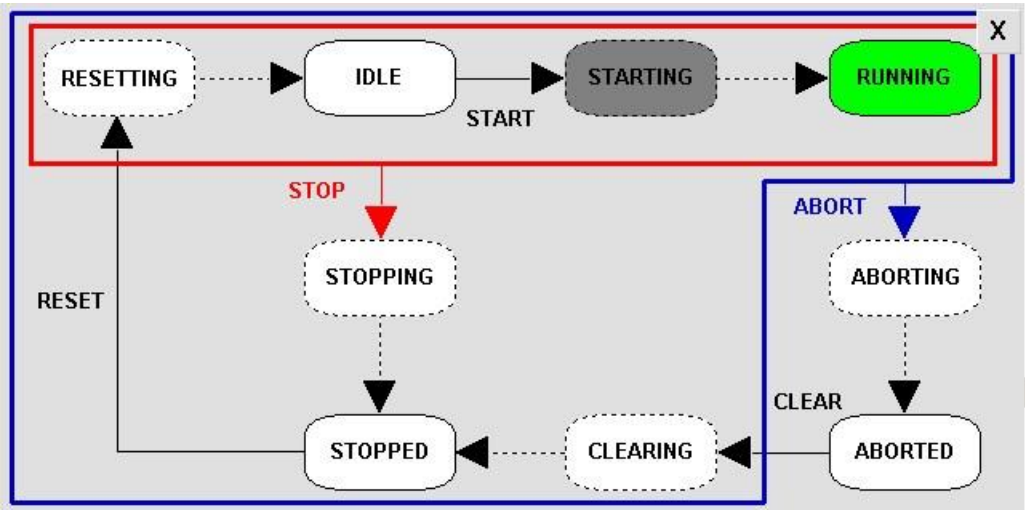


The **Axis Status** display lets you view general motion, axis, and drive status.

12. Press the button to see more status indicator. When you are done with the Axis Faceplate, close it by pressing the **[X]** in the top-right corner.

State Diagram Faceplate

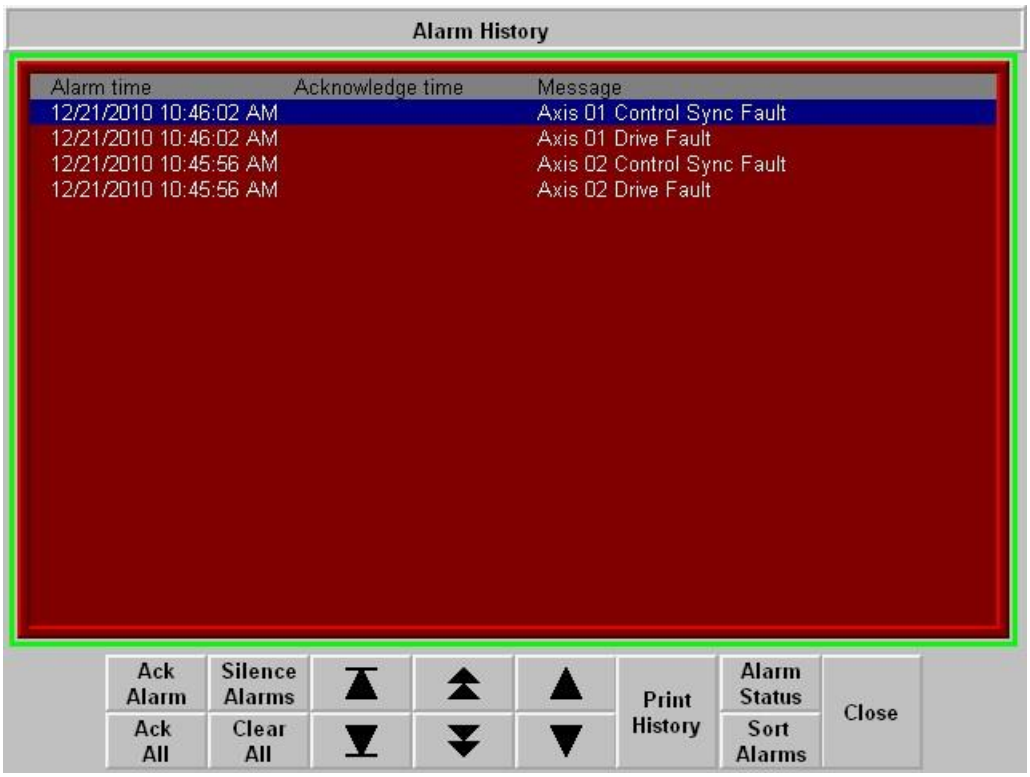
The State Diagram faceplate provides a graphical representation of the state machine. The **green** indicates the current state, while the **gray** indicates the previous machine state.



The State Diagram faceplate provides a quick reference for machine operators summarizing the relationship between machine states. When you are done with the State Diagram faceplate, close it by pressing the **[X]** in the top-right corner.

Alarm History Faceplate

The Alarm History faceplate provides a summary of current and past alarms for all the configured devices or drives configured in the application. The faceplate receives fault information directly from each of the device modules and applies a timestamp based on the order in which it was received.



The Alarm History faceplate can be an effective diagnostic tool for troubleshooting, helping machine operators pinpoint root causes quickly. When you are done with the Alarm History faceplate, close it by pressing the **Close** button on the bottom of the screen.

Equipment Status Faceplate

The Equipment Status faceplate lets you quickly load and configure a summary display of preconfigured status and diagnostic displays (faceplates). The Equipment Status faceplate works in conjunction with individual device faceplates and provides a single summary display of all the devices that may be configured for an application.

You can configure up to nine device faceplates to run with the Equipment Status screen and each device faceplate can be launched directly from it.

13. When you are done with the Equipment Status faceplate, close it by pressing the [X] in the top-right corner.

14. Click **Stop** to stop the drives.

Continue on to Lab 5.

Lab 5: Troubleshooting Techniques (Estimated Time 10 Minutes)

In this lab you will learn some basic troubleshooting techniques. In this lab, you will be asked to troubleshoot a Module Connection Fault using Logix Designer.

Diagnostic Capabilities of Logix Designer

First let's look at Logix Designer diagnostic capabilities using the file you saved in the previous lab.

1. Maximize Logix Designer. You should be **Online** with the controller.

2. From the **Controller**… folder in the Controller Organizer, right-click on **Controller Tags** and select **Monitor**

Tags…

3. Verify that you are on the Monitor Tags tab of the Controller Tags window...

4. Locate the **Axis01** (Data Type: **AXIS_CIP_DRIVE**) tag…

Name	Value	Force Mask	Style	Data Type
Axis01	{...}	{...}		AXIS_CIP_DRIVE
Axis01_Ctrl	{...}	{...}		UDT_ServoCtrl
Axis01_FP	{...}	{...}		CIP_Faceplate_AOI
Axis01_Module	{...}	{...}		UDT_ModuleStatus

Most of the diagnostic tags are automatically generated as part of the axis structure when an axis is created in Logix Designer.

5. Click the **[+]** to expand the tag to view the data structure.

6. Take a moment to scroll through and examine the **AXIS_CIP_DRIVE** axis structure.

> The tags are sorted alphabetically rather than by logical groupings. This can be switched by pressing the button in the Name header. If the logical groupings sort is used instead, the next few steps will have a different screen image.
>
> *'AXIS_CIP_DRIVE'* axis structure is significantly different than that of an *'AXIS_SERVO_DRIVE'*, which is used for SERCOS based servo drives. Some of the tags match and have an analogous function; other tags were added to the *'AXIS_CIP_DRIVE'* axis structure. **For comparison purposes, *'AXIS_SERVO_DRIVE'* axis structure contains 207 tags, while *'AXIS_CIP_DRIVE'* contains 463 tags.**

7. Locate the **Axis01.AxisFault** tag…

Name	Value	Force Mask	Style	Data Type
Axis01	{...}	{...}		AXIS_CIP_DRIVE
Axis01.AxisFault	16#0000_0000		Hex	DINT
Axis01.PhysicalAxisFault	0		Decimal	BOOL
Axis01.ModuleFault	0		Decimal	BOOL
Axis01.ConfigFault	0		Decimal	BOOL
Axis01.GroupFault	0		Decimal	BOOL
Axis01.MotionFault	0		Decimal	BOOL
Axis01.GuardFault	0		Decimal	BOOL
Axis01.InitializationFault	0		Decimal	BOOL
Axis01.APRFault	0		Decimal	BOOL

Notice the basic fault type bits are listed under the AxisFault word; when any fault condition is detected, the associated fault type bit is set.

8. Locate the **Axis01.ModuleFaults** tag (this is a different tag than **Axis01.ModuleFault**)…

Name	Value	Force Mask	Style	Data Type
Axis01.ModuleFaults	16#0000_0000		Hex	DINT
Axis01.ControlSyncFault	0		Decimal	BOOL
Axis01.ModuleSyncFault	0		Decimal	BOOL
Axis01.TimerEventFault	0		Decimal	BOOL
Axis01.ModuleHardwareFault	0		Decimal	BOOL
Axis01.ModuleConnFault	0		Decimal	BOOL
Axis01.ConnFormatFault	0		Decimal	BOOL
Axis01.LocalModeFault	0		Decimal	BOOL
Axis01.CPUWatchdogFault	0		Decimal	BOOL
Axis01.ClockJitterFault	0		Decimal	BOOL
Axis01.CyclicReadFault	0		Decimal	BOOL
Axis01.CyclicWriteFault	0		Decimal	BOOL
Axis01.ClockSkewFault	0		Decimal	BOOL
Axis01.ControlConnFault	0		Decimal	BOOL
Axis01.ControlClockSyncFault	0		Decimal	BOOL
Axis01.ModuleClockSyncFault	0		Decimal	BOOL
Axis01.LogicWatchdogFault	0		Decimal	BOOL
Axis01.DuplicateAddressFault	0		Decimal	BOOL
Axis01.ModuleAlarmStatus	16#0000_0000		Hex	DINT
Axis01.ControlSyncAlarm	0		Decimal	BOOL
Axis01.ModuleSyncAlarm	0		Decimal	BOOL
Axis01.TimerEventAlarm	0		Decimal	BOOL
Axis01.CPUOverloadAlarm	0		Decimal	BOOL
Axis01.ClockJitterAlarm	0		Decimal	BOOL
Axis01.OutOfRangeAlarm	0		Decimal	BOOL
Axis01.ClockSkewAlarm	0		Decimal	BOOL

Faults

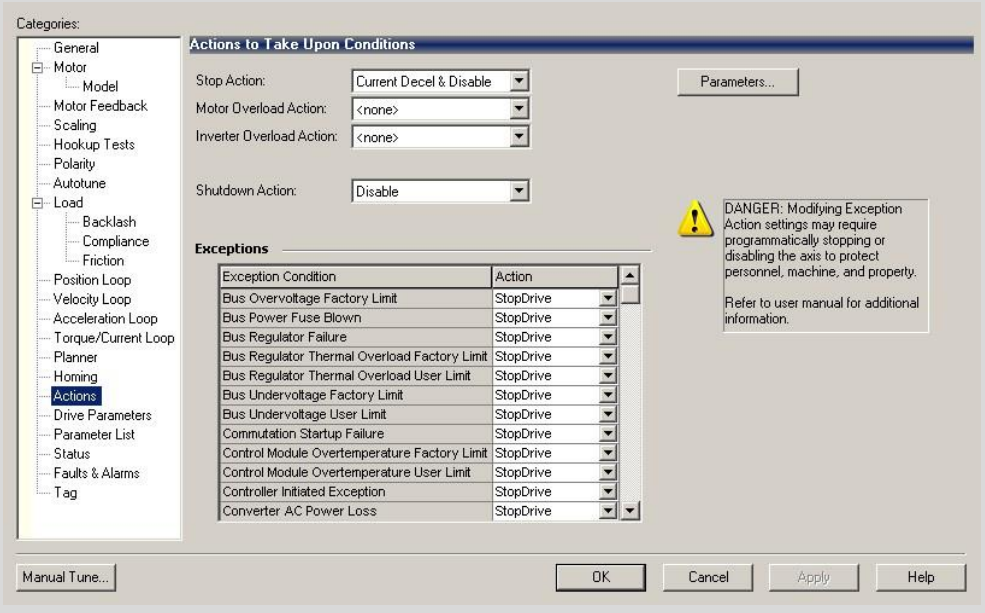
Alarms

Notice the fault types are further broken down into individual fault and alarm status bits.

This is one of the many benefits of the multi-discipline, integrated controller – you don't need to create code to collect motion controller diagnostics in the discrete controller or HMI.

Let's see how this works by generating a module fault condition.

The exception actions are used to set how an axis responds to different types of faults. The exception actions are located on the Actions page of the Axis Properties.



9. Disconnect the Ethernet cable between that runs between the drives.

Notice that after a few moments, **Axis01.ControlSyncFault** and **Axis01.ModuleConnFault** tags both register a value of **1**.

Name	Value	Force Mask	Style	Data Type
Axis01.ModuleFaults	16#0000_0041		Hex	DINT
Axis01.ControlSyncFault	1		Decimal	BOOL
Axis01.ModuleSyncFault	0		Decimal	BOOL
Axis01.TimerEventFault	0		Decimal	BOOL
Axis01.ModuleHardwareFault	0		Decimal	BOOL
Axis01.ModuleConnFault	1		Decimal	BOOL
Axis01.ConnFormatFault	0		Decimal	BOOL
Axis01.LocalModeFault	0		Decimal	BOOL

Note: Both faults indicate a loss of communications…

Control Sync Fault – Several consecutive updates from the controller have been lost.

Module Connection Fault – Communication with the controller has been lost.

10. From the **Motion Groups** > **MotionGroup** folder in the Controller Organizer, select **Axis01**…

2198-H003-ERS Drive01
Axis 1 - 192.168.1.24

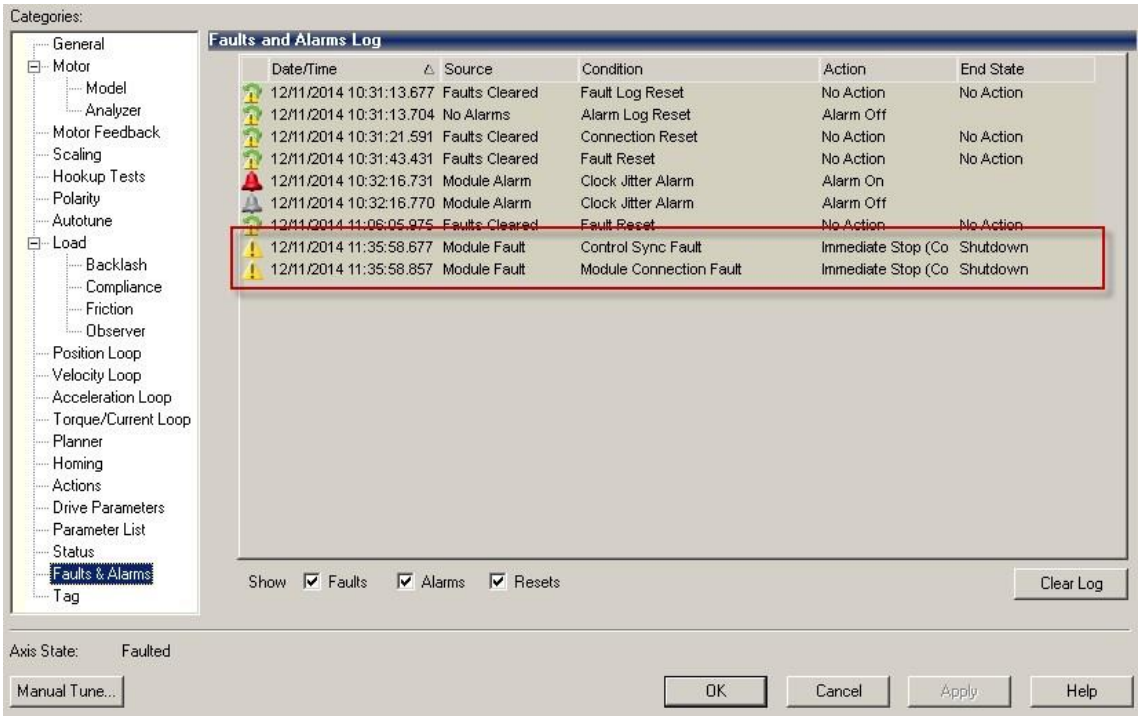
Type	AXIS_CIP_DRIVE
Description	
Axis State	Faulted
Axis Fault	ModuleFault
Module Faults	ControlSyncFault, ModuleConnFault
Group Fault	No Faults
Motion Fault	No Faults
Initialization Fault	No Faults
Attribute Error	No Faults
Guard Fault	No Faults
Start Inhibited	Not Inhibited
Motor Catalog	VPL-A1001M-P

Notice that both module faults are displayed in the quick view pane.

11. From the **Motion Groups** > **MotionGroup** folder in the Controller Organizer, right-click on **Axis01** and selected **Properties**…

The Axis Properties window pops up.

12. Navigate to the **Faults & Alarms** page…



Notice both module faults, plus additional information (Date/Time, etc.) are displayed. The Faults and Alarms Log was added to RSLogix 5000 v18 to support CIP Motion drives.

The 'Faults & Alarms' page displays the current state of both faults and alarms log structures currently in the controller for an axis.

The display is read-only except for the ability to clear logs independently.

The grid only shows entries when you are online with a controller.

When online, check or uncheck boxes in the Show row to toggle between showing and hiding the specified group of entries. Note that only the last 25 faults and alarms are displayed.

13. Click **Cancel** to close the Axis Properties window.

14. Lastly, notice that the Kinetix 5500 drive on the left is displaying a fault message on the display and the **Module** status indicator should be **flashing** red.

15. Reconnect the Ethernet cable.
    Verify that after a few moments, **Axis01.ControlSyncFault** and **Axis01.ModuleConnFault** tags both returned to a value of **0** and the drive is now displaying **"STOPPPED"** again.

16. It may take up to a minute for the drive to reconnect to the controller. Each time the drive is connected to a controller, it is reinitialized.

APPENDIX (**_Optional_**)

The following appendices are **OPTIONAL** and provides some examples of more advanced motion topics. Even the most experienced motion control engineers occasionally struggle with complex applications. The following appendices will cover advanced topics such as finding an optimal tradeoff between response and stability when tuning, CAM instructions, drive multiplexing and more. Come along and learn practical solutions to getting that machine really flying!

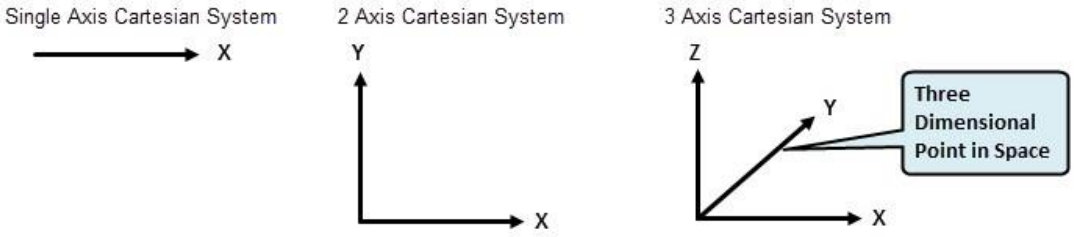
The motion advanced topics lab consists of a variety of labs that will introduce you to motion examples and programming features. The intent of the labs is to expand your knowledge of detailed motion topics by providing a simplified example of use and function of advanced concepts.

Each lab should take approximately 20-30 minutes to complete. Choose the labs topics that most interest you so you can complete them in the allotted session time. The PF527 is NOT used in these advanced motion topics.

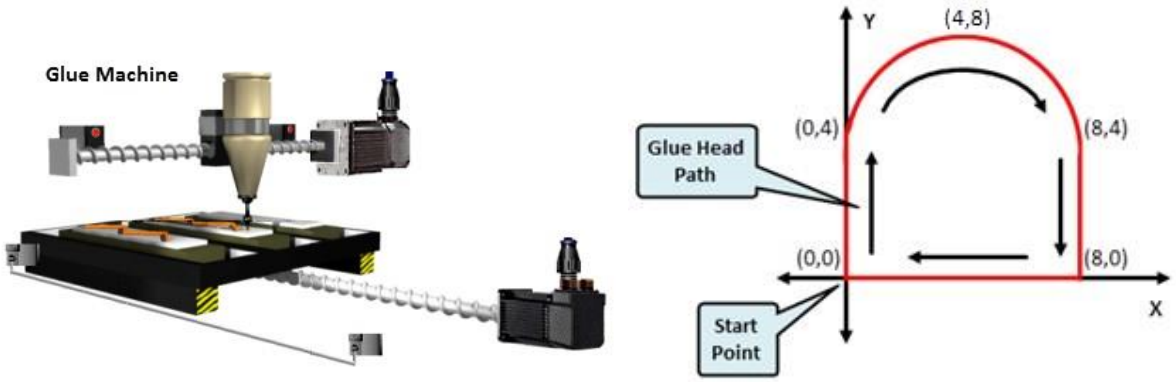
Appendix A: Logix Coordinated Motion

In this lab we will create a coordinate system in the Motion Group and demonstrate the Logix Coordinated Motion instructions.

The multi-axis coordinated motion instructions are used to perform linear and circular moves in single and multidimensional spaces. A Cartesian coordinate system in Logix can include one, two or three axis



Let's look at an example of a two axis Cartesian system application. Most motion applications require multiple motion moves to be executed in succession. A gluing machine is a typical example. The simulated gluing machine will apply a glue bead following the tool path shown below.



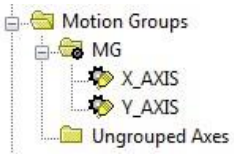
This application can be accomplished with the following:

- (3) MCLM Instructions
- (2) MCCM Instructions
- A simple ladder based state machine

Creating the Coordinate System.

1. Open file **\Desktop\Lab Files\ Coordinated Motion Lab\CIP_XY_CoordMotion_Begin.ACD**

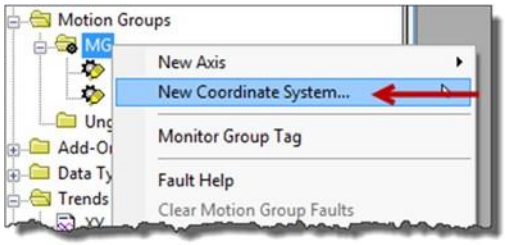
2. In the Controller Organizer, note that under the Motion Group (MG) two CIP Drive Axes have been created and configured.



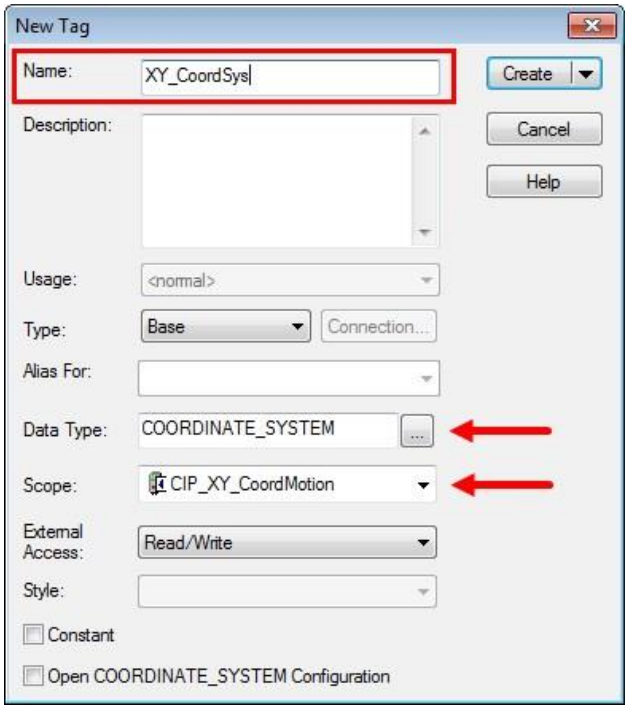
At this point, we could program basic motion instructions such as MAJ, MAM, MAG, etc. To program coordinated motion, however we need to create a Coordinate System tag under the motion group.

> **Note:** The maximum number of axes that can be associated with one Coordinate System is limited to three axes.

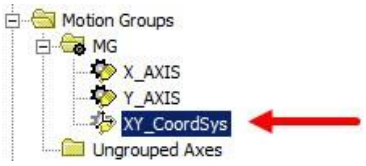
3. Right-click on the Motion Group (MG) and select **New Coordinate System…**



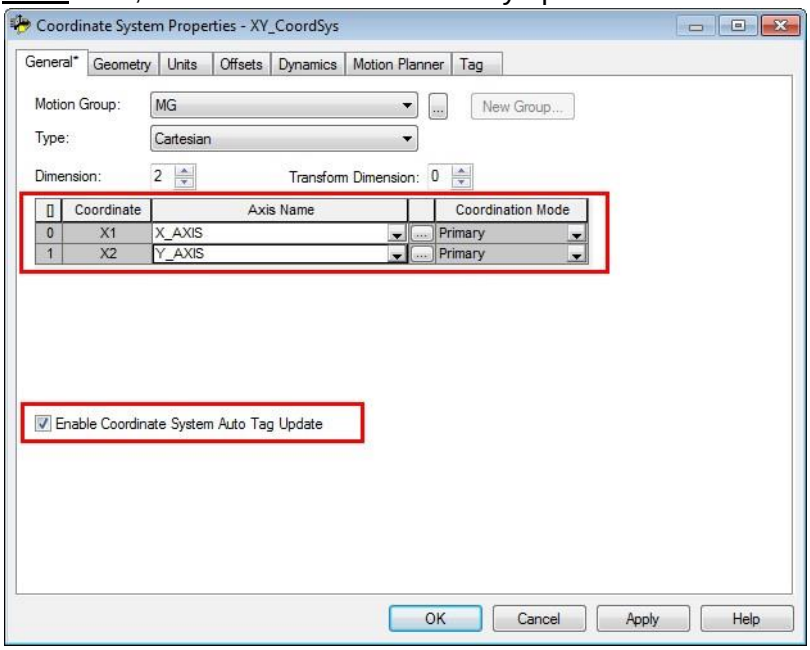
4. Enter "**XY_CoordSys**" for the name. Verify the Data Type COORDINATE_SYSTEM is automatically set and the Scope is controller scoped. Click **Create** when done.



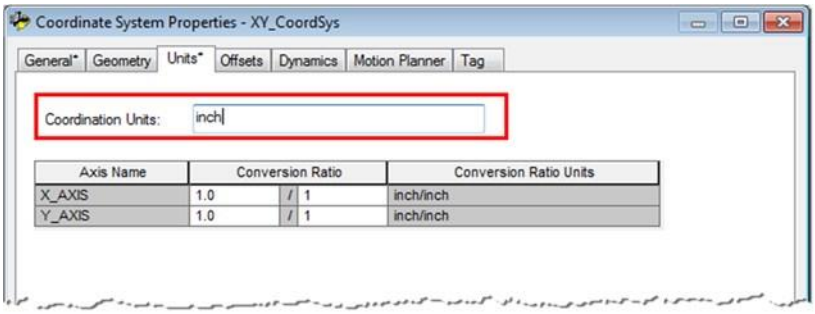
5. Let's configure our newly created coordinate system. Double click on **XY_CoordSys** under the motion group to open the configuration dialog window.



6. Assign Coordinate X1 to **X_AXIS** and X2 to **Y_AXIS**. Also verify that the Enable Coordinate System Auto Tag Update check box is checked. We are going to use these values in our trend later, so we want them continually updated.

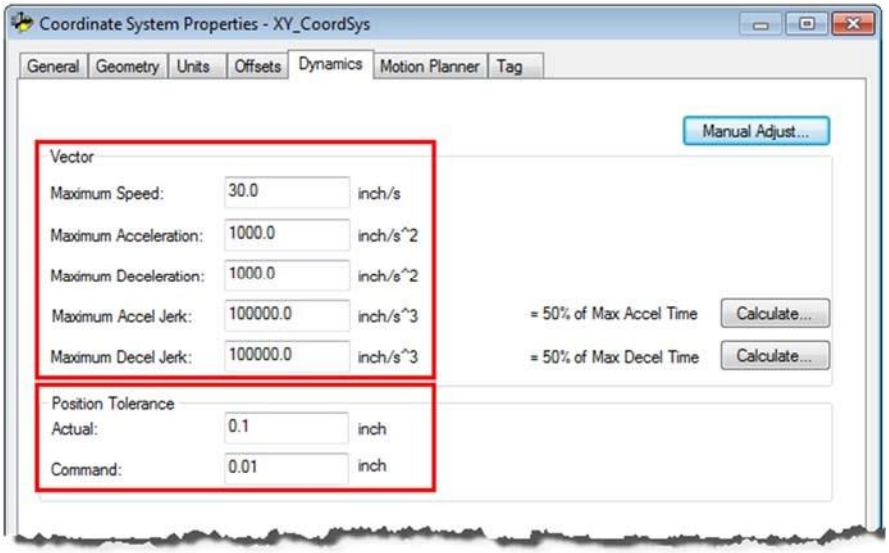


7. Select the Units tab and enter "**inch**" for Coordination Units.


Coordinate System Properties - XY_CoordSys
General* Geometry Units* Offsets Dynamics Motion Planner Tag
Coordination Units: inch
Axis Name
Conversion Ratio
Conversion Ratio Units
X_AXIS 1.0 / 1 inch/inch
Y_AXIS 1.0 / 1 inch/inch

In this lab, the specified Units for the axes and Coordination Units are the same, so the Conversion Ratio Units shown will be inch/inch and the ratio will be 1/1. The option to fill in a Conversion Ratio is more useful when we are dealing with different units. For example, if the axes units were in Degrees then the Conversion Ratio Units column would display Degrees/Inch.

8. Select the **Dynamics** tab and enter the data into each field as detailed in the screen shot below.

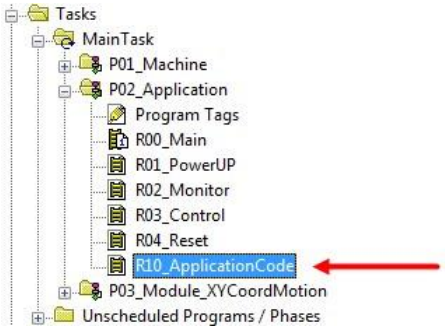


Note: Each axis has its own Dynamics (defined during individual axis configuration), but so does the Coordinate System. It has its own Vector Max Speed, Accel and Decel, and Accel & Decel Jerk as defined here.

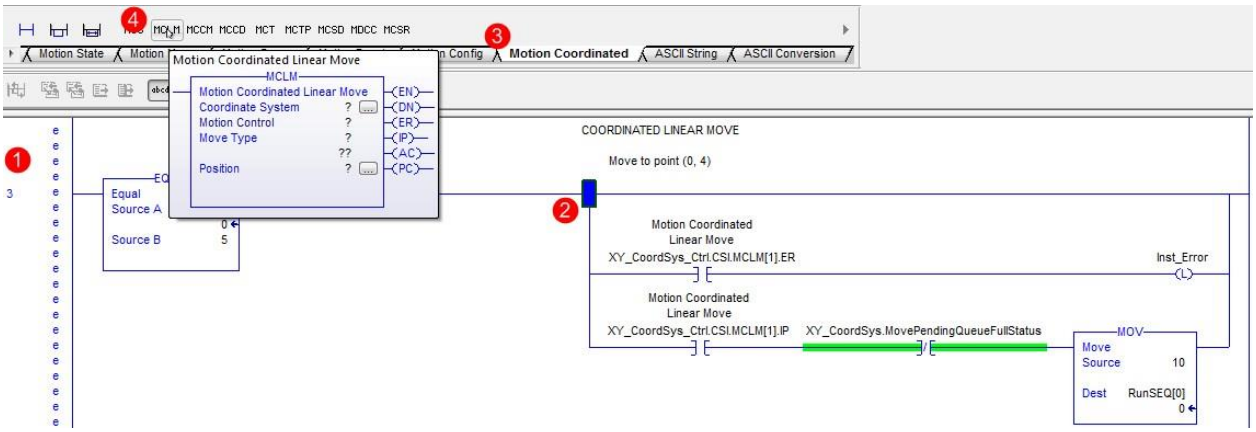
9. Click **OK** to save your changes and close the dialog window.

Add a Motion Coordinated Linear Move (MCLM) and a Motion Coordinated Circular Move (MCCM)

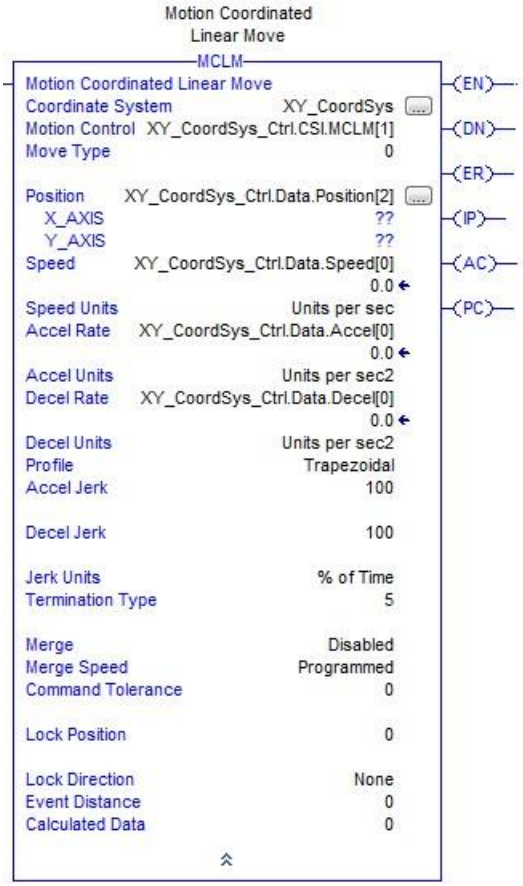
10. From the Controller Organizer navigate to the routine **Main Task > P02_Application > R10_ApplicationCode** and open it. This routine contains our application specific code and will be used throughout this lab.



11. Add a **MCLM** instruction by …
    1. Go to rung 3
    2. Place your curser here
    3. Select **Motion Coordinated** instruction tab
    4. Select the **MCLM** instruction



12. Enter instruction data as shown in the picture to the right. It is important that you select the correct tag for each entry.



Coordinate System: **XY_CoordSys**

Motion Control: **XY_CoordSys_Ctrl.CSI.MCLM[1]**

Move Type: **0**

Position: **XY_CoordSys_Ctrl.Data.Position[2]**

Speed: **XY_CoordSys_Ctrl.Data.Speed[0]**

Speed Units: **Units per sec**

Accel Rate: **XY_CoordSys_Ctrl.Data.Accel[0]**

Accel Units: **Units per sec2**

Decel Rate: **XY_CoordSys_Ctrl.Data.Decel[0]**

Decel Units: **Units per sec2**

Profile: **Trapezoidal**

Accel and Decel Jerk: **100**

Jerk Units: **% of Time**

Termination Type: **5**

Merge: **Disabled**

Merge Speed: **Programmed**

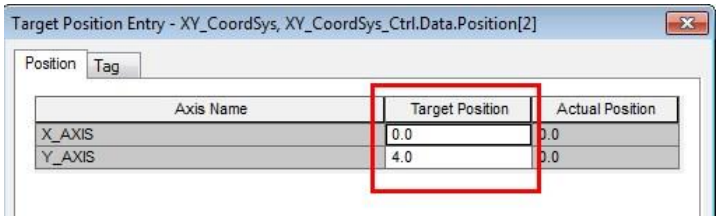
Command Tolerance and Lock Position: **0** for both

Lock Direction: **None**

Event Distance and Calculated Data: **0** for both

13. After completing the instruction entries, click on the **ellipsis** next to the **Position tag** to open the dialog window.

14. Enter the **Target Position** values, **X_AXIS=0.0 and Y_AXIS=4.0**, for this MCLM move. Then click **OK**.



15. Add a **Motion Coordinated Circular Move (MCCM)** instruction on rung 4 (similar process as step 11).

16. Enter instruction data for the MCCM as shown in the picture to the right. It is important that you select the correct tag for each entry.

Coordinate System: **XY_CoordSys**

Motion Control: **XY_CoordSys_Ctrl.CSI.MCCM[0]**

Move Type: **0**

Position: **XY_CoordSys_Ctrl.Data.Position[4]**

Circle Type: **1**

Via/Center/Radius: **XY_CoordSys_Ctrl.Data.ViaCenterRadius[0]**

Direction: **0**

Speed: **XY_CoordSys_Ctrl.Data.Speed[0]**

Speed Units: **Units per sec**

Accel Rate: **XY_CoordSys_Ctrl.Data.Accel[0]**

Accel Units: **Units per sec2**

Decel Rate: **XY_CoordSys_Ctrl.Data.Decel[0]**

Decel Units: **Units per sec2**

Profile: **Trapezoidal**

Accel and Decel Jerk: **100**

Jerk Units: **% of Time**

Termination Type: **5**

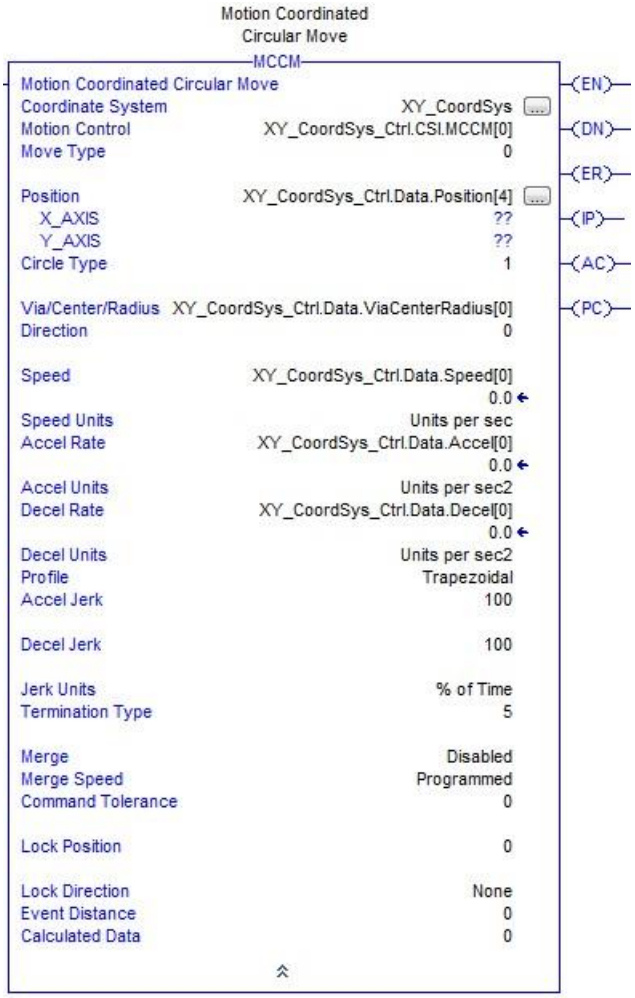
Merge: **Disabled**

Merge Speed: **Programmed**

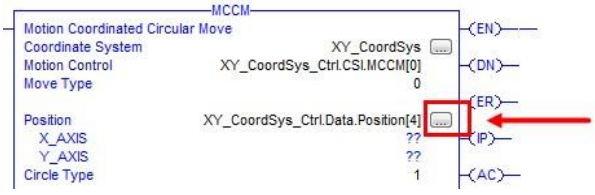
Command Tolerance and Lock Position: **0** for both

Lock Direction: **None**

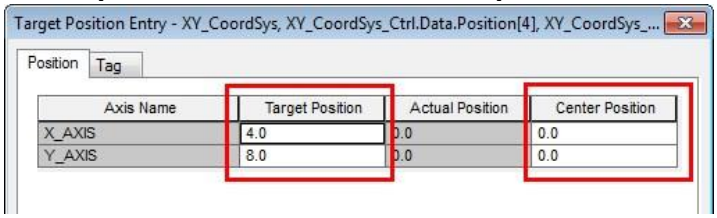
Event Distance and Calculated Data: **0** for both



17. Open the Position dialog window for the MCCM instruction.



18. Enter the **Target Position** values **(X_AXIS=4.0 and Y_AXIS=8.0)** and the **Center Position** values **(X_AXIS=0.0 and Y_AXIS=0.0)** for this MCCM move.

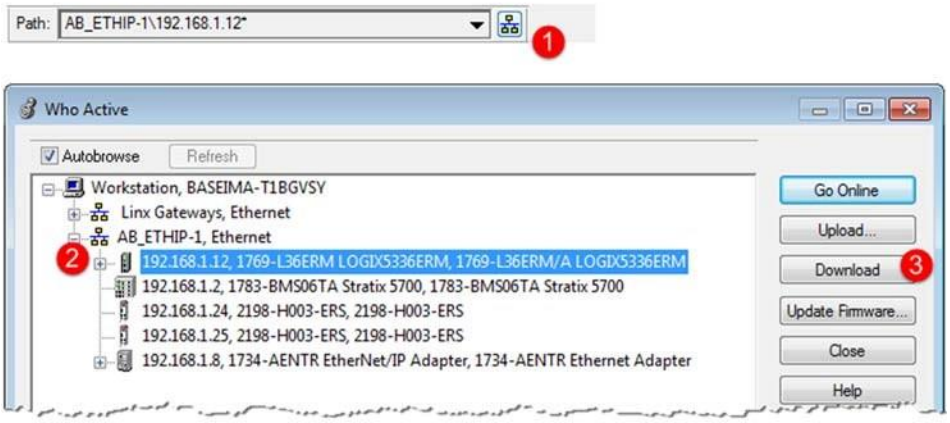


19. Click **OK** to lock in the values.

20. Save your program.

Execute the Coordinated Motion Profile

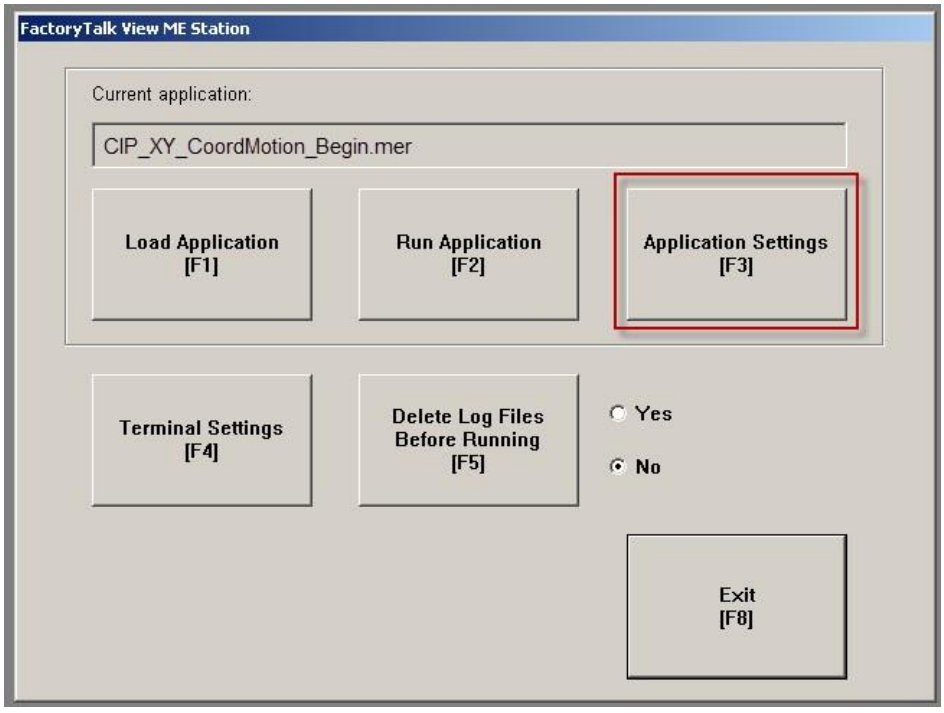
21. Download the program to the controller at **192.168.1.12**.
    1. Click **'Who Active'** button.
    2. Select controller from the Ethernet driver.
    3. Click **Download** button.
    4. On pop-up dialog window, click the **Download** button.



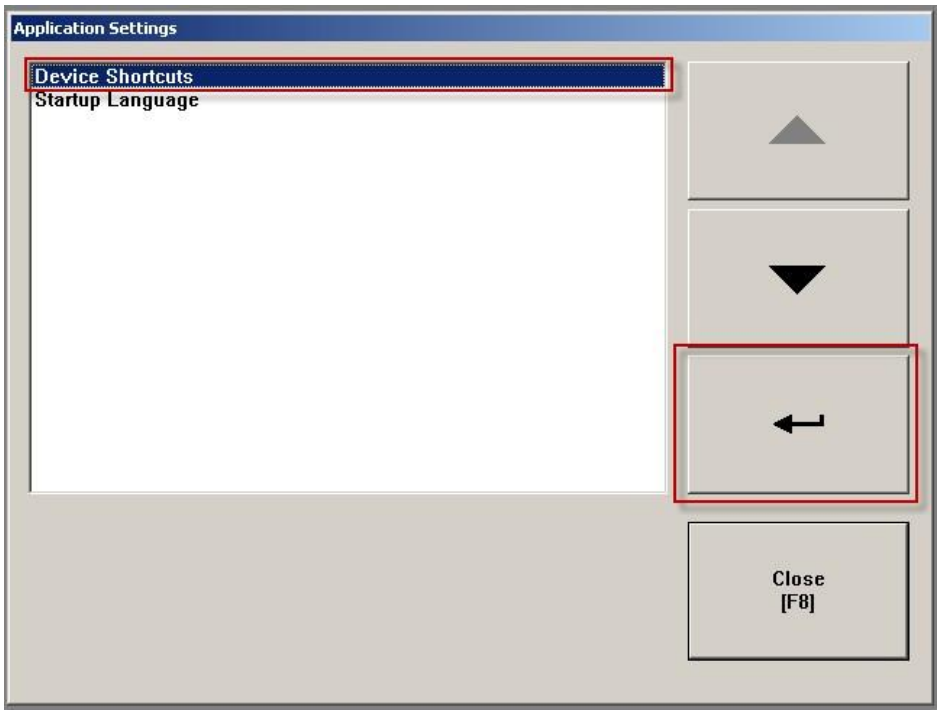
22. Once the program is downloaded, set the controller to **Run Mode**.
23. Open file **\Desktop\Lab Files\Coordinated Motion Lab\CIP_XY_CoordMotion_Begin.mer**, by double-clicking it, to run the lab HMI.

24. Click **Yes** button on the 'Replace Local System Directory' dialog.

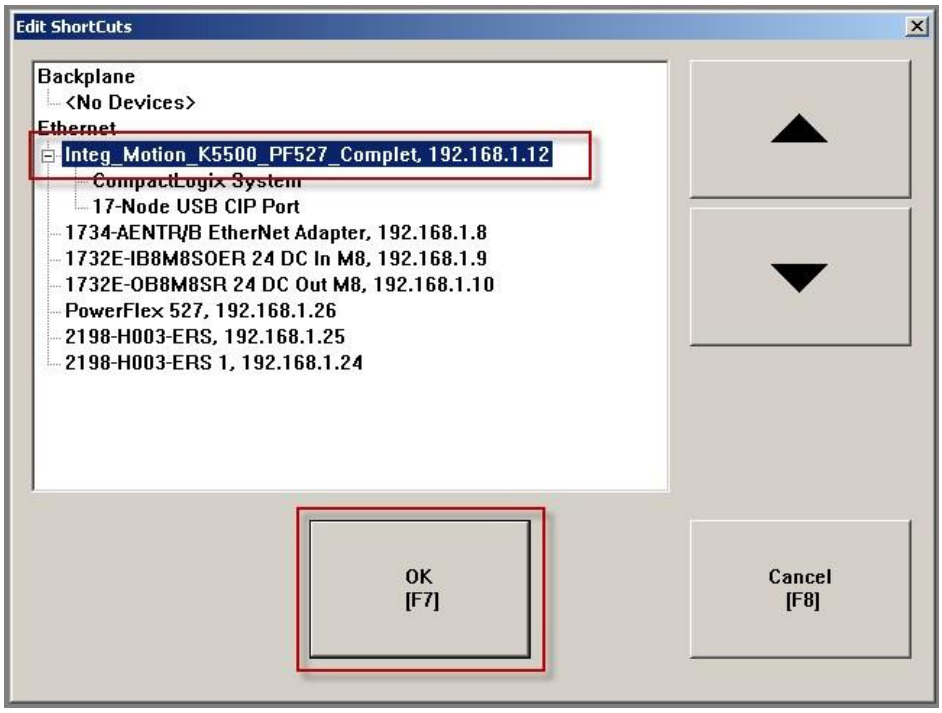
25. Click **Application Settings**



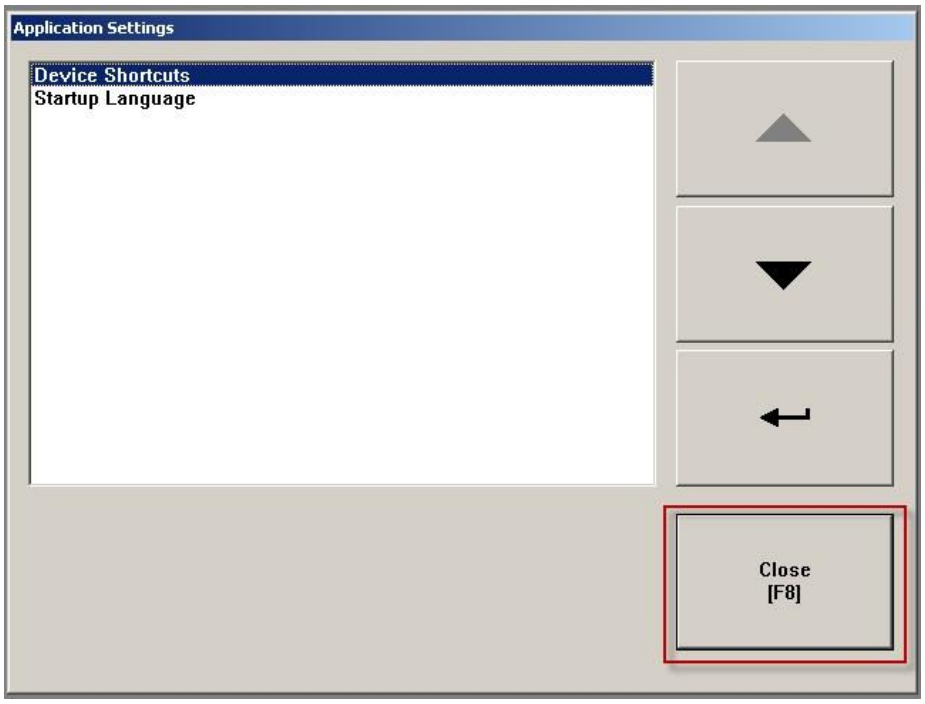
26. Select **Device Shortcuts** and hit the Enter button.



27. Verify that the controller at **192.168.1.12** IP address is selected as shown below. Click **OK**.



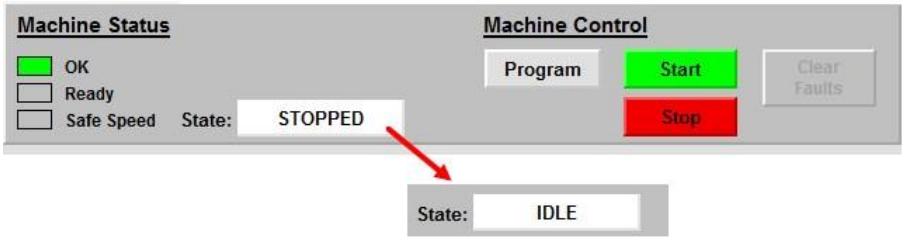
28. Click the **Close** button.



29. Click Run Application button on the ‘FactoryTalk View ME Station’ dialog.

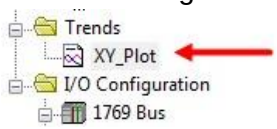
30. On the HMI press the **‘Clear Faults’** button to reset the system.

31. Now press the **'Start'** button to ready the system for motion. The Machine State should transition from 'Stopped' to 'Idle'.

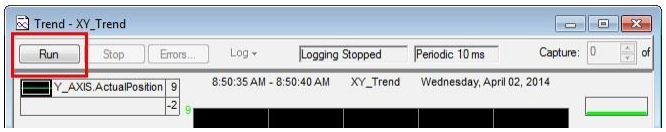


32. Press the **'Start'** button again to start executing the motion application code. Both motors in the demo box should now be rotating per the program motion instructions.

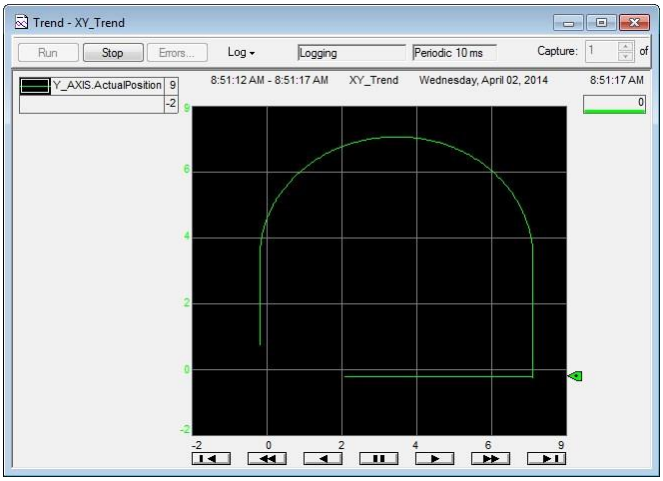
33. From the Controller Organizer expand the **Trends** folder and open the trend **XY_Plot** by double-clicking.



34. Click **Run** to begin trending the X and Y axes.



35. Your trend should look like the one pictured here.



This concludes this lab.

Appendix B: Master Driven Speed Control (MDAC) Lab

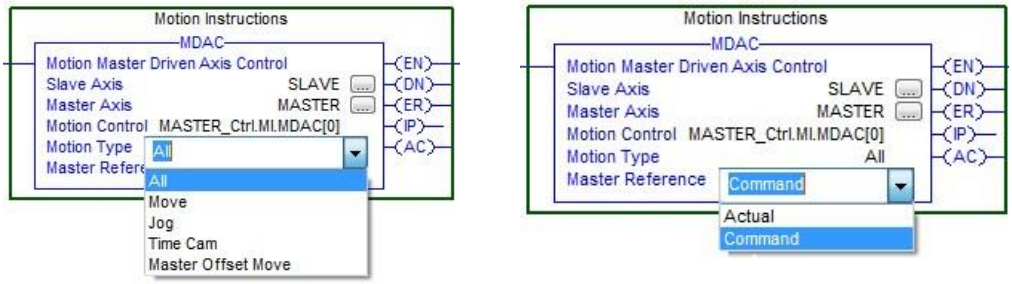
RSLogix 5000 V20 introduced a feature called Master Driven Speed Control (MDSC). The concept of this feature is to ***create a motion driven speed control system where the slave's position is based on the master's position***.

To define the relationship between the master and slave axes, two new instructions have been created:

- ***The MDAC (Motion Master Driven Axis Control) - single axis***
- ***The MDCC (Motion Master Driven Coordinated Control) - coordinated axes***

The slave's speed can be directly proportional to the master's speed like a gear ratio. Or the slave's speed can be in master units like a single entry position cam. This MDSC feature is now an option in the MAJ, MAM, MCLM, MCCM and the MATC motion instructions.

The MDAC has two pull down menus for Motion Type and Master Reference. Motion Type allows you to filter the type of moves that will use the MDSC feature. And the Master Reference allows you to select either Command or Actual Position.



Within the motion move instructions, 4 new instruction operands have been introduced:

- ***Lock Position*** □ ***Event Distance***
- ***Lock Direction*** □ ***Calculated Data***

And for instruction unit operands we have introduced new values of; ***Units per MasterUnit***, ***Master Units*** and ***Seconds***.

For the MAM instruction we have the added feature of programming the move in time (time as the master unit). You define the end point of the move and the total time of the move.

In this lab we will cover some of the basic functionality of Motion Drive Speed Control by doing the following:

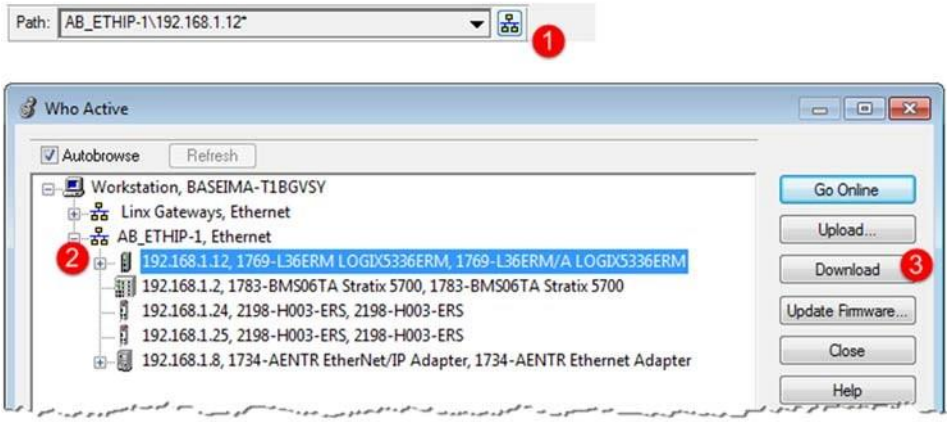
- Use the MAM in classic (time driven) mode with speed, accel, decel and jerk defined in user units
- Use of the MDAC instruction
- Use the MAM in MDSC mode with speed, accel, decel and jerk defined in Units of MasterUnits

Execute the MAM Instruction in Classic Mode

Here we will show the basic use of the MAM instruction in classic (time driven) mode with speed, accel, decel and jerk defined in user units. It will give a point of reference for using the MDCS functionality.

1. Open file **\Desktop\Lab Files\Master Driven Speed Control Lab\MDSC_MAM_Begin.ACD**.

2. Download the program to the controller at **192.168.1.12**.
    1. Click **'Who Active'** button.
    2. Select controller from the Ethernet driver.
    3. Click **Download** button.
    4. On pop-up dialog window, click the **Download** button.

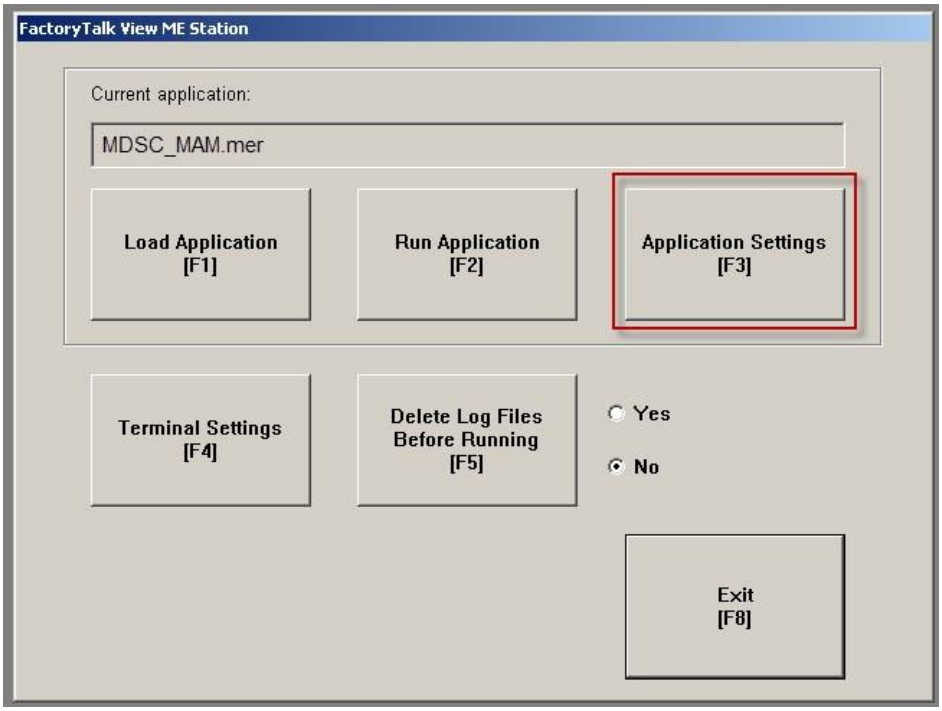


3. Set the controller to **Run Mode**.

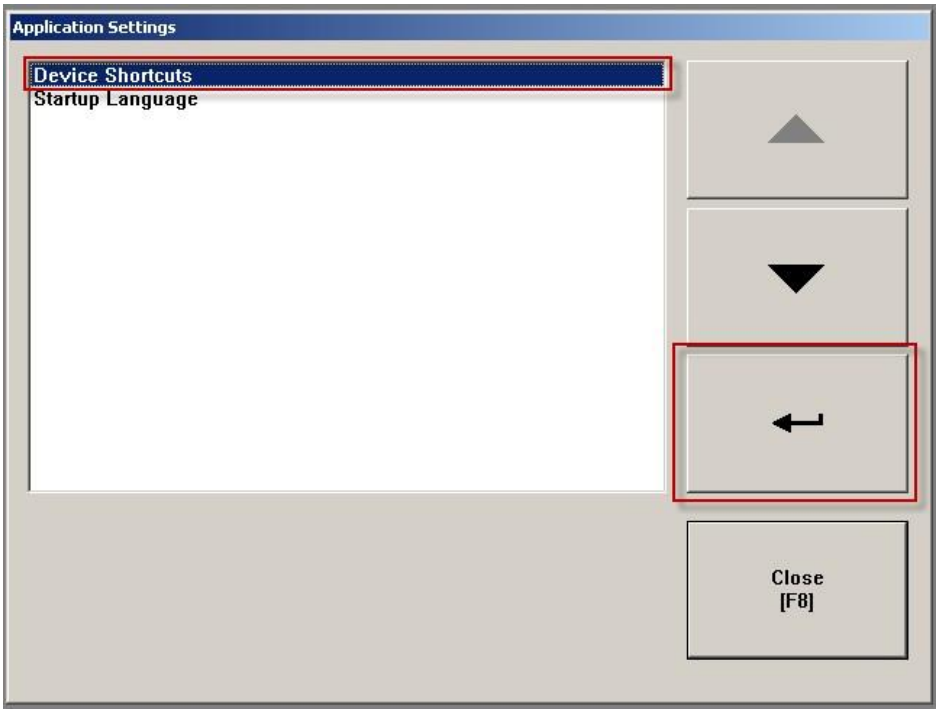
4. Open file **\Desktop\Lab Files\Master Driven Speed Control Lab\MDSC_MAM.mer**, by double-clicking it, to run the lab HMI.

5. On the **'Replace Local System Directory'** dialog, press the **Yes** button to continue.

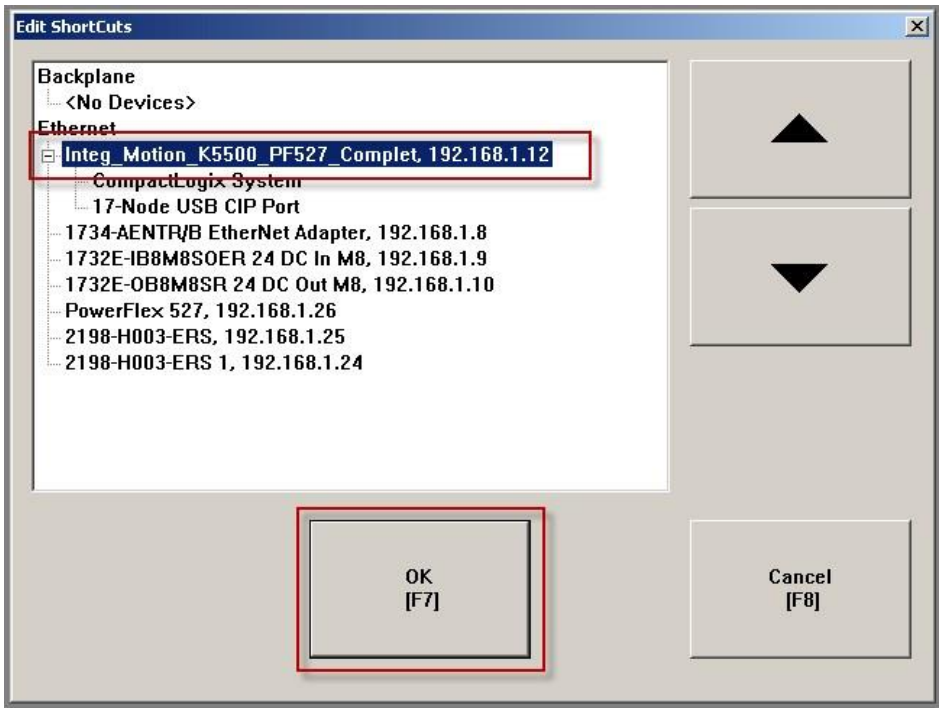
6. Click **Application Settings**



7. Select **Device Shortcuts** and hit the Enter button.



8. Verify that the controller at **192.168.1.12** IP address is selected as shown below. Click **OK**.

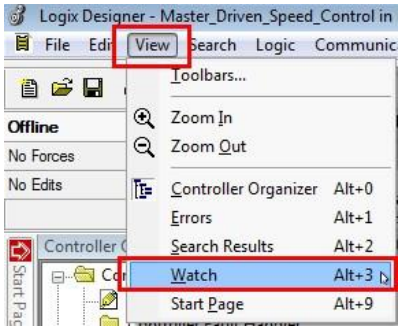


9. Click the **Close** button.

10. On the **'FactoryTalk View ME Station'** dialog, press the **Run Application** button.

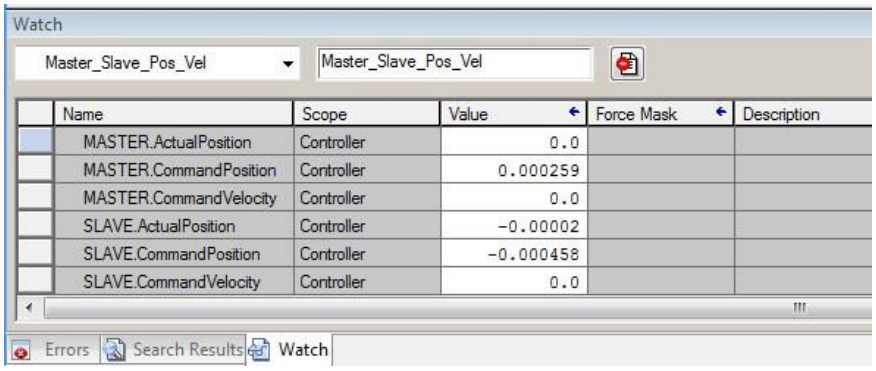
11. On the HMI press the **'Clear Faults'** button to reset the system.

12. Go back to Studio 5000 and from the main menu select ***View > Watch***.

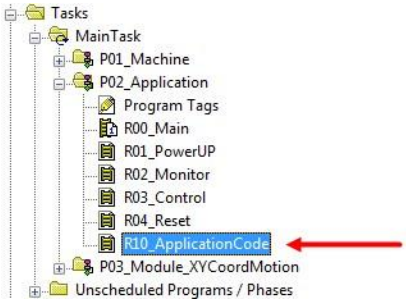


13. In the Watch window use the pull down for **Current Routine** and select **Master_Slave_Pos_Vel**.

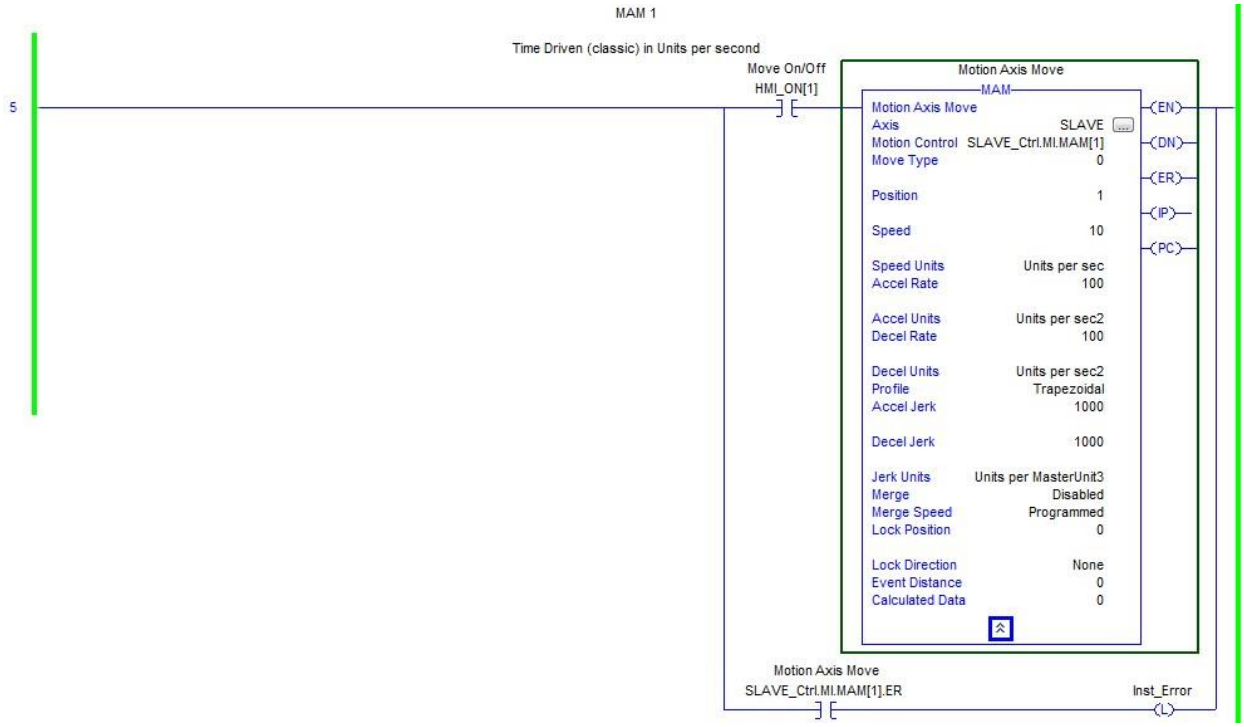
14. On the HMI push the **Enable/Home** button to enable and home the axes. Notice the values in the watch window.



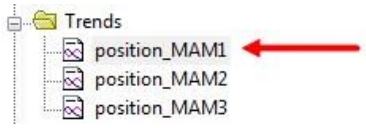
15. From the Controller Organizer navigate to the routine **Main Task > P02_Application > R10_ApplicationCode** and open it. This routine contains our application specific code and will be used throughout this lab.



16. We will start with executing the MAM in rung 5. This is a very conservative move; going from position 0 to 1 at 10 units/sec (10 revs/sec). This is known as time driven or classic mode.



17. We want to observe the move in a trend. Under **Trends** in the **Controller Organizer**, double-click on the trend **position_MAM1** to open it.



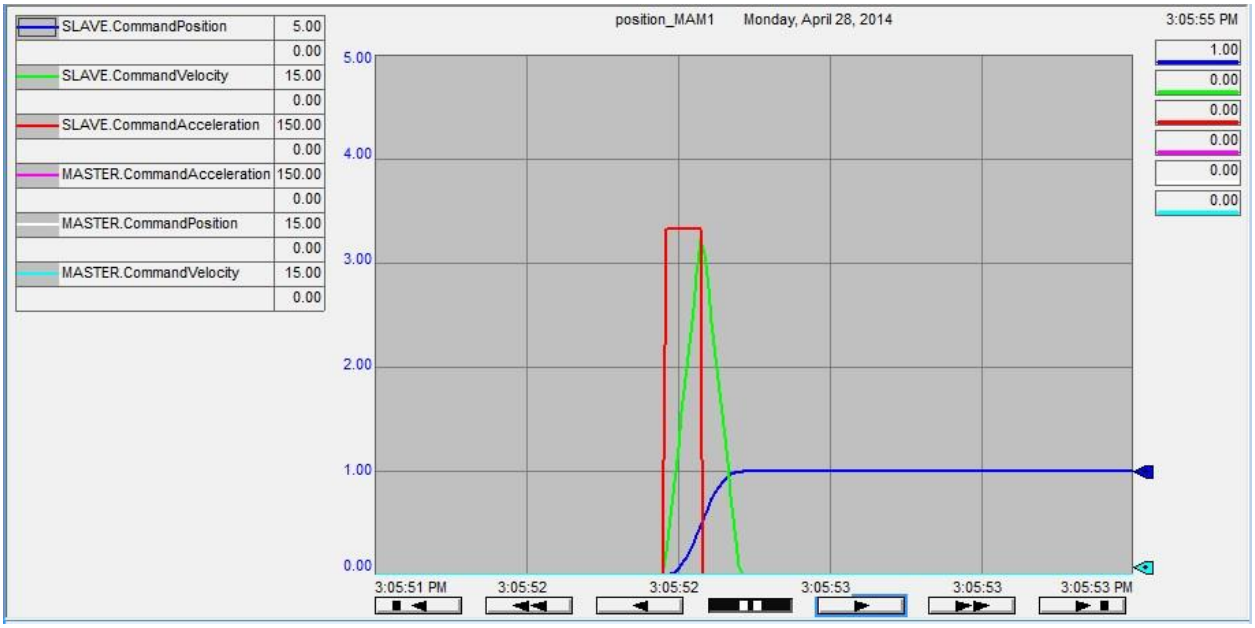
18. Click **Run** to start collecting data in the trend.

19. On the HMI, push the **MAM1** button to execute the move. We don't need the MDAC instruction for a MAM running in time driven (classic) mode. Note the motion of the right motor in the demo box.

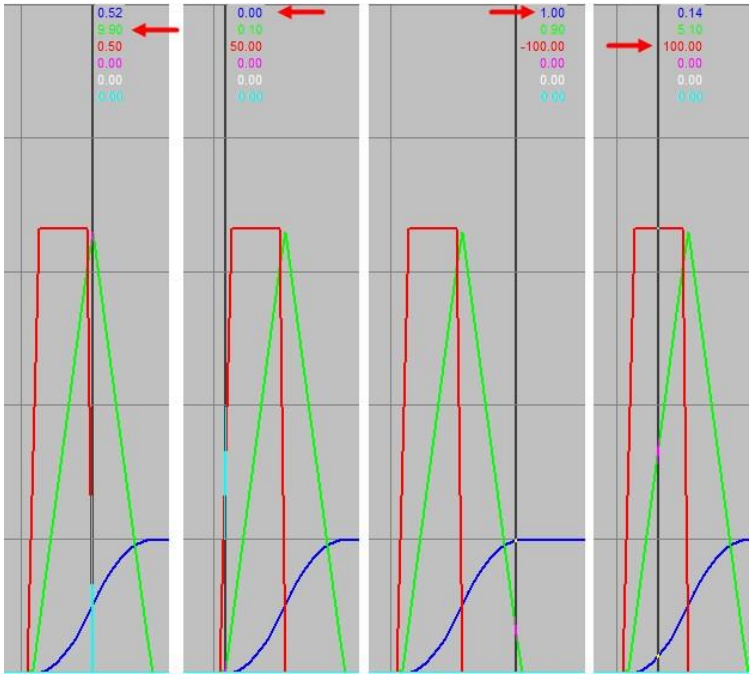
20. After the move completes, click the **Stop** button in the trend.

21. Use the scroll back in time buttons (below the graph) until you see the MAM in the trend.

    Your trend should look similar to this …



22. **Click on the peak of the velocity curve** (green triangle) and verify that the top speed of 10rev/sec was reached in the move. **Click on the start and end of the position curve** (blue) to verify the start and end positions of 0 and 1. **Click in the middle of the acceleration curve** (red) to verify the maximum acceleration of 100rev/sec^2.



Execute and Verify a MAM in MDSC mode using Units per MasterUnit

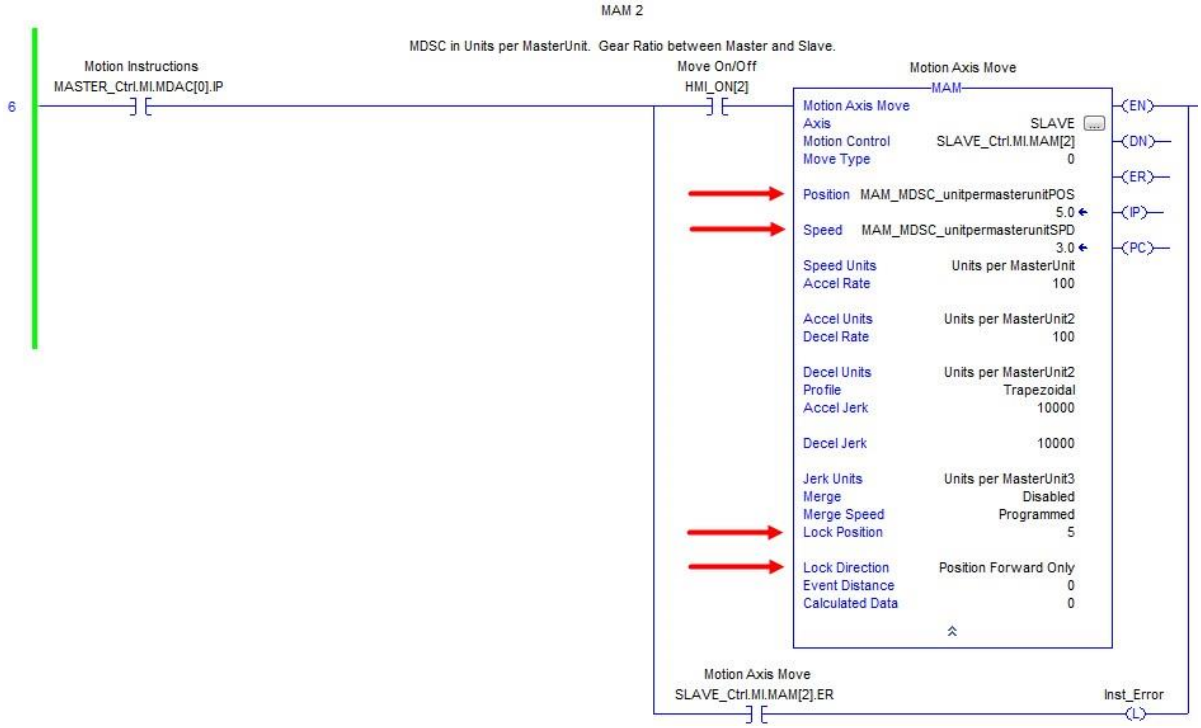
Now we will trigger a slave MAM based on a master's position and set the speed as a gear ratio to the master. The MAM will be programmed in *Units per Master Unit*.

To use the MAM in MDSC Mode we need to …

- Set up the relationship between the Master and Slave axes with a MDAC instruction

- Set your MAM units for speed, accel/decel and jerk to *Units per Master Units*
- Select a Lock Position of the Master that will trigger the Slave to move
- Select a Lock Direction of the Master, the direction that along with the Lock Position will trigger the Slave to move.

1. Rung 6 contains our move instruction. Notice that we have the condition that the MDAC instruction needs to be IP (in process) before we can execute the move, otherwise the move instruction would error.



The MAM instruction will execute an absolute move from position 0 to position 5 (5revs) with a speed of 3.0 Units per MasterUnit. The Master (on the left) will be jogging at 1 unit/sec (1rev/sec). The MAM will execute when the rung goes from false to true and the Master passes the Lock Position of 5 and will execute in a Lock Direction of Position Forward Only.

2. **Close** any open Trends.

3. **Open** the trend called **position_MAM2**. Click **Run** to start collecting data in the trend.

4. **Press** the **Enable/Home** button on the HMI to enable the drives and set both axes to position 0.

5. You will have less than 5 seconds to complete step 7 once you complete step 6. This is because after 5 seconds the Master axis will have passed the 5 revs position where the Slave move is to be triggered.

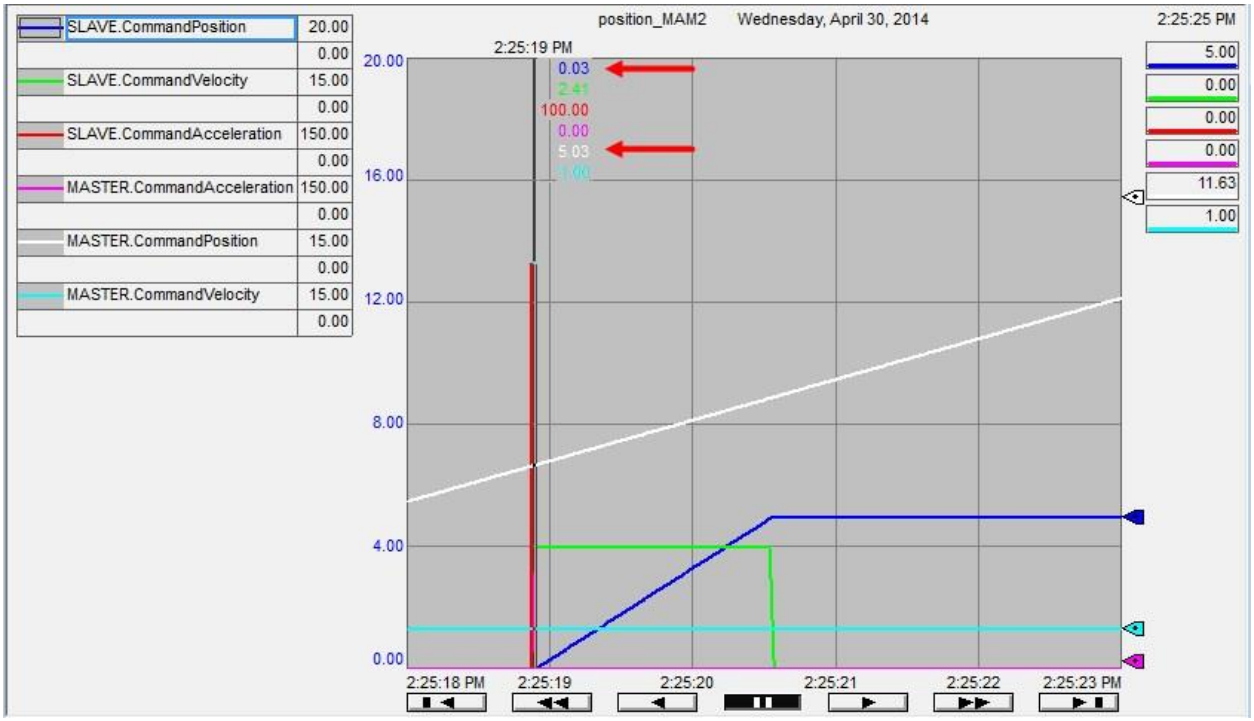
6. **Press** the **MDSC** button on the HMI. You should see the MDAC instruction on rung 2 go IP (in process).

7. **Press** the **MAM 2** button on the HMI.

8. After the Slave axis (right motor) move completes, click the **Stop** button in the trend.
9. **Press** the **Stop** button on the HMI to stop the Master axis.

10. Use the **scroll back in time** buttons until you see the Slave's move in the trend.

11. **Click on the left edge of the green velocity curve**. Verify that the Active Value Bar shows that the Slave's motion started when the Master's position was 5. Verify that the starting Slave's position was 0.



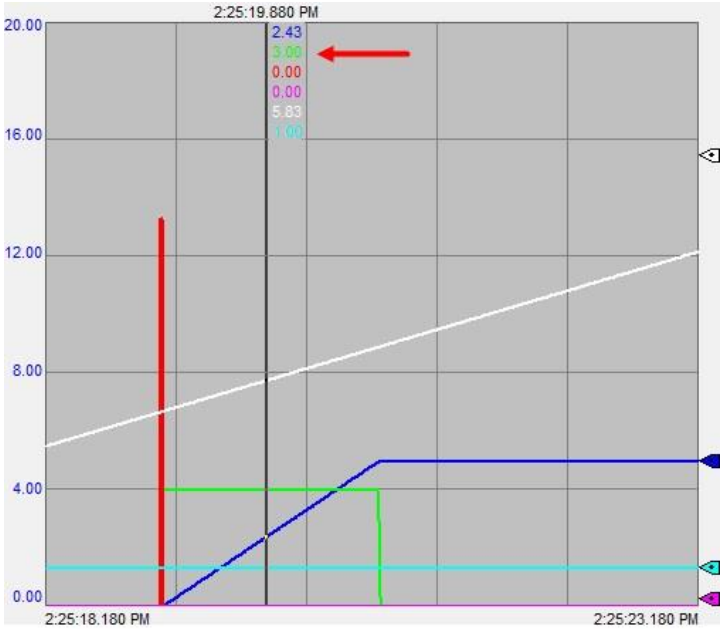
12. Let's calculate the total time of the Slave's move and compare it to the values captured in the trend. We asked the Slave to move 5 position units at 3 times the Master's speed. The Master is running at 1 position units/sec. Notice how Units per MasterUnit is like a gear ratio. So solving for total time of the move we have:

5 position units / 3 position units/sec = 1.667 sec

13. Click on the trend at the start and end of the Slave's move. Note the time of each. Then calculate the difference in time. Is it approximately 1.667 seconds?

14. Here it is in this example …. 2:25:20.760 – 2:25:19.080 = 1.680 seconds

15. Note the velocity of the move by clicking in the middle of the green velocity curve. Is it 3 rev/sec or 3 times the Master's speed of 1 rev/sec? Yes it is. ***Wow, moves with a built in gear ratio … Nice!***



16. Let's try another MAM with MDSC in Units per MasterUnits.

17. Change the **Master jog speed** in the **MAJ** instruction on **rung 3** to a value of **2**.

18. Change the **Slave position** to **10** in the **MAM** on **rung 6**. Leave the speed at 3.

19. Click **Run** in **position_MAM2** to restart the trend.

20. Press the **Enable/Home** button on the HMI.

21. Press the **MDSC** button on the HMI.

22. Press the **MAM 2** button on the HMI.

23. After the Slave's move completes, click the **Stop** button in the trend.

24. Press the **Stop** button on the HMI to stop the Master axis.

25. Again verify that the Slave's motion started when the Master's position was 5. Also, check that the starting Slave's position was 0 and the final position is 10.

26. Click in the middle of the Slave velocity curve (green) and verify that the value is 6.00. So our gear ratio for speed is …

    3 * 2 rev/sec or 6 revs/sec

27. Now let's verify the total time of the Slave's move. So solving for total time of the move we have …

    10 position units / 6 position units/sec = 1.667 sec

    Click on the trend at the start and end of the Slave's move. Note the time of each. Then calculate the difference in time. Is it approximately 1.667 seconds?

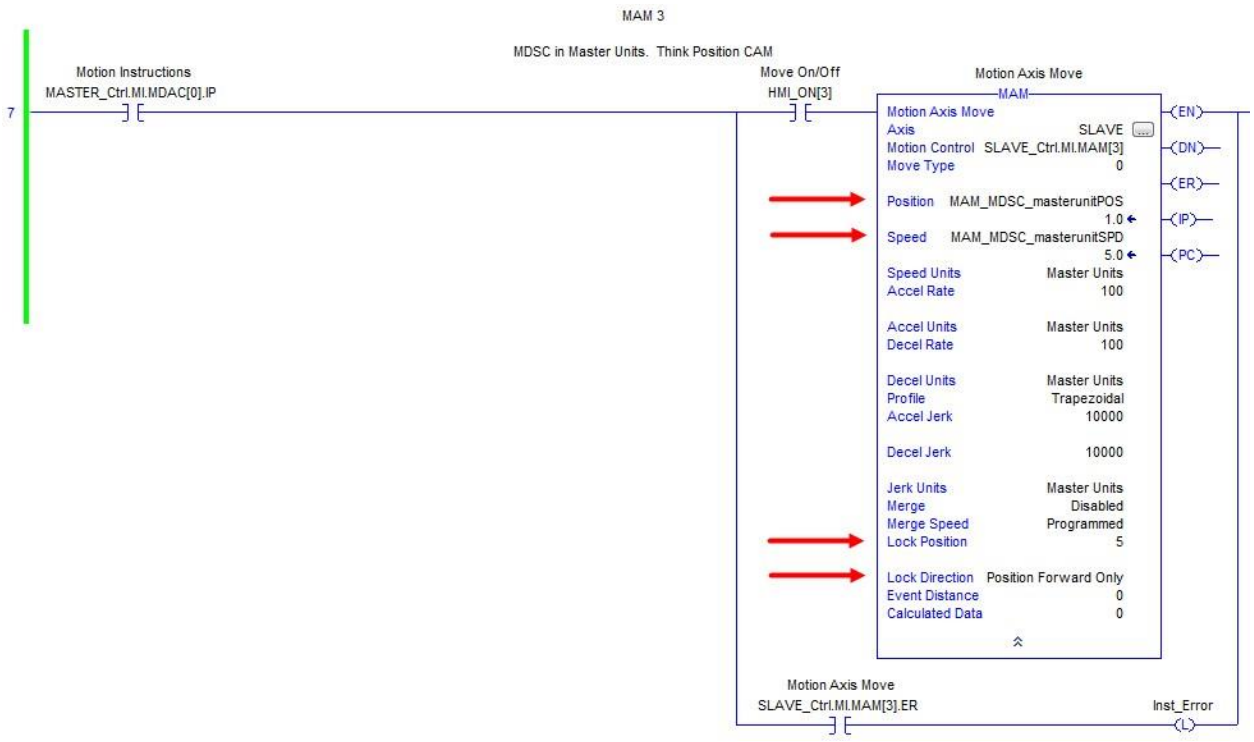
Execute and Verify a MAM in MDSC mode using MasterUnits

We just saw how you can set up the MAM where the speed of the move is geared to the Master. How about having the speed of the Slave, be in Master Units, so the instruction behaves more like a one shot position cam.

Same steps as the previous section, but now the MAM units for speed, accel/decel and jerk need to be set to *Master Units*.

1. To start, change the **Master jog speed** back to **1rev/sec** on rung 3.

2. Rung 7 contains our move instruction. In this MAM the lock position of the Master is still 5. Here we want to move the Slave 1 position unit (1rev) in the time it takes for the Master to go 5 position units (revs). This will be a slow move.



3. **Close** any open Trends.

4. **Open** the trend called **position_MAM3**. Click **Run** to start collecting data in the trend.

5. **Press** the **Enable/Home** button on the HMI to enable the drives and set both axes to position 0.

6. Again you will have less than 5 seconds to complete step 8 once you complete step 7. This is because after 5 seconds the Master axis will have passed the 5 revs position where the Slave move is to be triggered.

7. **Press** the **MDSC** button on the HMI. You should see the MDAC instruction go IP (in process).

8. **Press** the **MAM 3** button on the HMI.

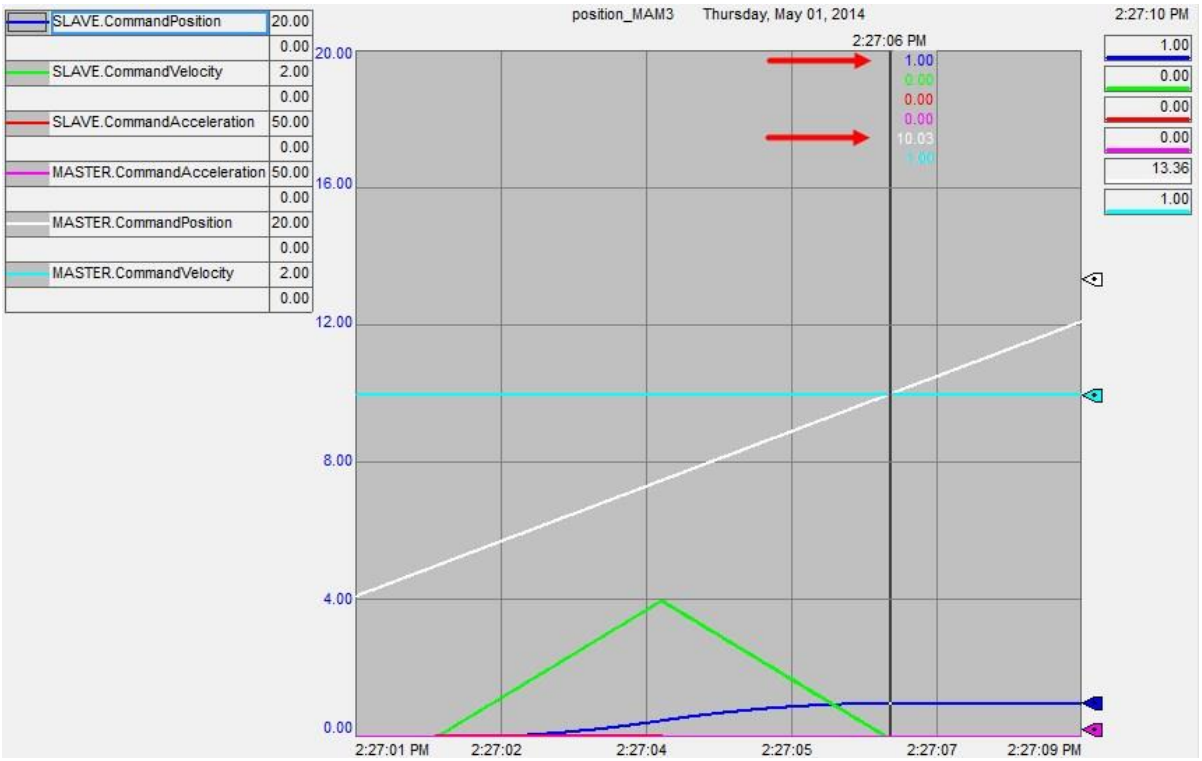
9. After the Slave axis (right motor) move completes, click the **Stop** button in the trend.

10. **Press** the **Stop** button on the HMI to stop the Master axis.

11. Use the **scroll back in time** buttons until you see the Slave's move in the trend.

12. **Click on the right edge of the green velocity curve**. Verify that the Active Value Bar shows that the Master's position at the end of the Slave's move is 10 … {5revs (lock position) + 5revs (master units) = 10}.

Investigate the trend until you are convinced that the Slave moved 1 position unit in the time it took the Master to move 5 position units.



13. Let's try another MAM with MDSC in MasterUnits.

14. Change the **Position** to **25** in the **MAM** on **rung 7**. Change the **Speed** to **7**.

15. Click **Run** in **position_MAM3** to restart the trend.

16. Press the **Enable/Home** button on the HMI.

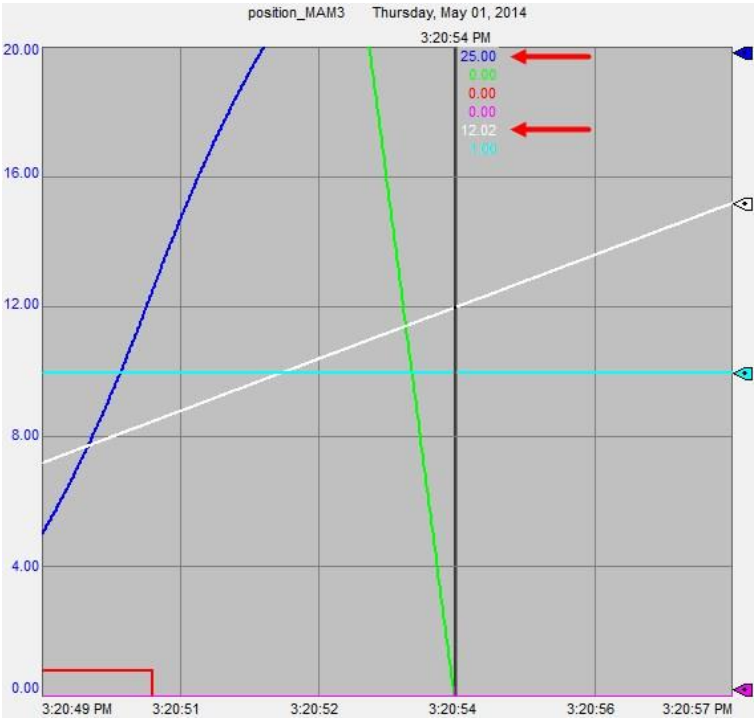
17. Press the **MDSC** button on the HMI.

18. Press the **MAM 3** button on the HMI.

19. After the Slave's move completes, click the **Stop** button in the trend.
20. Press the **Stop** button on the HMI to stop the Master axis.

21. What would you expect for the Master distance at the completion of the Slave move?

    Click on the right edge of the green velocity curve. Looks like the Slave's motion ended at 25revs when the Master's position was 12revs {5revs (lock position) + 7revs (master units) = 12}. ***This almost seems like a MAPC instruction where the CamProfile array contains only 1 value … Nice!***



22. Feel free to make other modifications to the values to drive home the operation of the MDSC motion feature in Logix.

This concludes this lab.

Appendix C: PCAM Rotary Knife Application Lab

In this lab, we illustrate the process of designing Cams to control the motion of an Example Machine. The topic of Cam Design can become complicated when it is necessary to achieve peak performance. However there are Basic Cam Designs that can be incorporated into the program that are quite useful. This lab focuses on the Basic Cam Designs that requires an understanding of motion control but does not require a motion control expert.

Based on the labs time constraints if you struggle to fully grasp the in-depth discussion continue to progress through the content.

The lab is structured to allow you to complete the lab and observe the results without fully understanding the in-depth discussion.

Overview of the Example Machine

The purpose of the example machine, shown in Figure 1, is to cut Product at the specific Cut Length. The topology of the machine includes two axes of motion. The first axis controls a Conveyor, referenced as the Master Axis, that carries Product at a constant speed. The second axis controls a Rotary Knife that cuts the Product, in the middle of the Cut Region at the Bottom Dead Center, at the specific Cut Length.

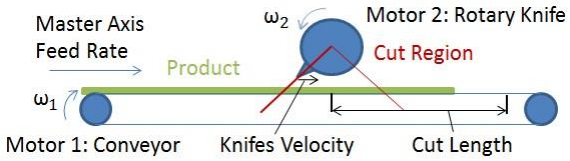


Figure 1 – Machine Topology

The radius of the Conveyors spindles are 5cm and Rotary Knife blade is 15cm. The Product Cut Length is 200cm. It can be assumed that the load torque of the Rotary Knife is much greater than the torque generated during the cutting operation. Thus the torque is primarily dependent on the Cam Profile and the Feed Rate. The goal is to maximize the production rate without exceeding a specified torque which can cause excessive wear. The exact design objectives are not specified for the lab. While operating the machine note the machine performance based on the Feed Rate and Cam Design. ***Overview of How to Control the Machine with a Basic Cam Design***

For the machine, described in Figure 1, there are several methods to control the axes in order to cut the Product at a specific length. Each approach has pros and cons in regards to the ease of use and the achievable Feed Rate. In this lab, we are going to develop a Basic Cam Profile to coordinate motion between the Conveyor and Rotary Knife. This approach will provide good machine performance and minimize the design complexity for a typical engineer.

Conveyor Control: This axis is defined as the Master Axis to control the machine. The machines Feed Rate is controlled by the speed of Motor 1. The exact relationship, based on the circumference of a spindle, is defined as

$$\frac{rev}{s}\Big] = \frac{Feed\ Rate\ \left[\frac{cm}{s}\right]}{2\pi\ r_{spindle}\left[\frac{cm}{rev}\right]}$$

Equation 1 – Relationship between Motor 1 Speed to Feed Rate

In this application the Axis controlling the Conveyor is scaled based on the length of Product that has run through the machine.

Rotary Knife Control: When controlling the example machine the obtainable Feed Rate is dictated by the methodology used to control the Rotary Knife. When using a Basic Cam Design to control the Rotary Knife the main objectives include

- Controlling the Knife's Linear Velocity, within the Cut Region, to match the machines Feed Rate
- Reorienting the Rotary Knife, during the Make-Up Move, to appropriately cut the Product

In this lab a 2-segment Cam Profile is used to control the orientation of the Rotary Knife based on the Master Axis. A 2-segment

Cam Profile simply refers to a single Cam Profile that consists of two parts, which are referenced as the Cut Region and the Make-Up Move. While the Knife Blade is in the Cut Region the inverse kinematics, that are equations that describes the Knifes linear velocity relative to the angular velocity, are used to maintain a constant linear velocity parallel to the Conveyor. As the Knife Blade exits the Cut Region the Make-Up Move reorients the Knife to cut the Product at the specific Cut Length. In order for the Knife Velocity to equal the Feed Rate, through the Cut Region, the Cam Profile must account for the Knifes Kinematics illustrated below.

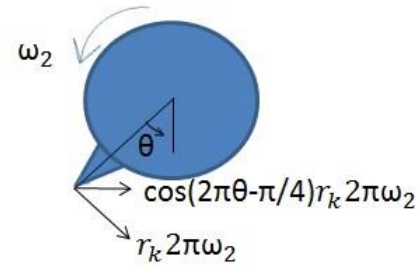


Figure 2 – Rotary Knife Kinematics

This may sound complicated but it is fairly straight forward. It is well understood that the linear velocity of the Knife Blade is equal to 2π where is the radius of the knife [cm] and the angular velocity [rev/s]. Thus by geometric identities the Knife Velocity is equal to $2\pi\omega\cos(2\pi\theta-\pi/4)$ where θ is the angular orientation [revs]. The zero angular orientation is shown in Figure 2 and the orientation ranges from 0 – 1. In this case 0 orientation was not chosen to be Bottom Dead Center (BDC) where the Knife Blade cuts the Product. Rather, it is chosen to be 45° from BDC which corresponds to the π/4 phase shift in the geometric identity. This offset is chosen to simplify the creation of the Cam Table in Excel and avoid a roll over condition. In order for the Knife's Velocity to match the Feed Rate, through the Cut Region, the angular velocity must be equal to the inverse kinematics shown in Equation 2. If you struggle to follow the geometry that computes the linear velocity, continue to progress through the lab content, or you can ask the lab instructor for more details.

$$\frac{rev}{s}\Big] = \frac{Feed\ Rate\left[\frac{cm}{s}\right]}{r_k[cm]2\pi\cos\left(2\pi\theta-\frac{\pi}{}\right)}$$

Equation 2 – Rotary Knife Inverse Kinematic

Once the objective of the Cam Design is understood the Cam Table is created which includes the Cutting Region and Make-Up Move. In this lab, for simplicity, the Cam Table is created using Excel to clearly illustrate the design process that includes:

- Labeling the configuration variables, including units, and loading the system parameters. This includes the Feed Rate
  [cm/s], Knife Radius [cm], and a Sample Rate [s]. Note: the Sample Rate dictates the number of points included in the Cam Table for the cutting region. In Studio 5000 arbitrarily increasing the number of Cam Points may not increase the performance. In fact at a certain point increasing the number of points can decrease the performance due to interpolation errors. The rule of thumb is to use eight Cam Points per acceleration ramp.

	E	F
1	feed rate [cm/s]	10
2	knife radius [cm]	15
3	sample rate [s]	0.10

- Labeling the system dependent variables, including units, and defining the initial values. This includes Master Axis [cm], θ [rev], and ω [rev/s]. The initial value of cell C2 is computed from Equations 2 with the formula F$1/(2*3.14*COS(2*3.14*B2-3.14/4)*F$2).

	A	B	C
1	Master Axis [cm]	θ [rev]	ω [rev/s]
2	0	0	0.150069088

- The next iteration of the dependent variables are computed as A3 = Master Axis = A2+F$1*F$3, B3 = θ = B2+C2*F$3, and C3 = ω = F$1/(2*3.14*COS(2*3.14*B3-3.14/4)*F$2). Subsequent iterations, rows, are filled based on the same equations until the orientation of the knife exits the Cut Region. In this case the orientation is equal to 90° or 0.25 revs. The Cam Profile for the Cut Region is shown in Figure 3. The figure below illustrates the Cam Table for the Cutting Region.

	A	B	C
1	Master Axis [cm]	θ [rev]	ω [rev/s]
2	0	0	0.150069088
3	1	0.015006909	0.137729691
4	2	0.028779878	0.129000304
5	3	0.041679908	0.122555016
6	4	0.05393541	0.117683439
7	5	0.065703754	0.113968037
8	6	0.077100558	0.111147962
9	7	0.088215354	0.109054002
10	8	0.099120754	0.107574689
11	9	0.109878223	0.106637596
12	10	0.120541982	0.106198729
13	11	0.131161855	0.106236643
14	12	0.14178552	0.10674966
15	13	0.152460486	0.10775545
16	14	0.163236031	0.109292844
17	15	0.174165315	0.111426276
18	16	0.185307943	0.114253883
19	17	0.196733331	0.117921439
20	18	0.208525475	0.122646376
21	19	0.220790112	0.128760708
22	20	0.233666183	0.13679221
23	21	0.247345404	0.147630547

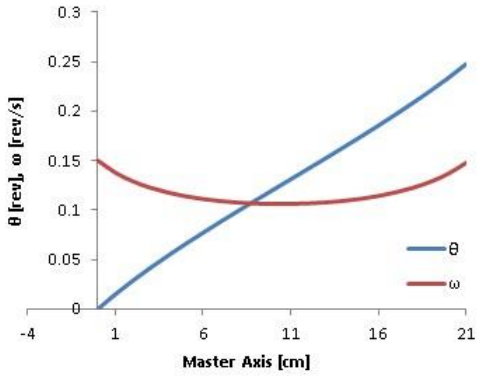


Figure 3 – Cut Regions Cam Profile

- As the Knife exits the Cut Region a Make-Up Move reorients the knife to appropriately cut the Product. This means the Knife is reoriented to 1 rev when the Master Axis reaches the Cut Length. The Make-Up Move impacts the obtainable performance and design complexity. In this lab a linear Make-Up Move, shown below, is used to illustrate the design procedure.

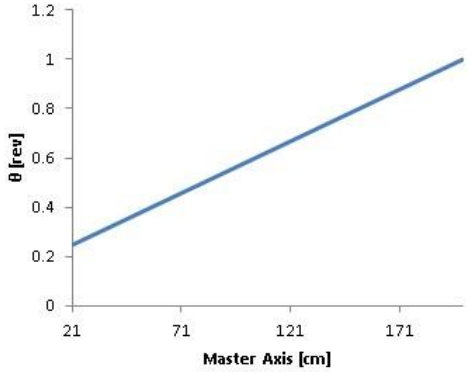
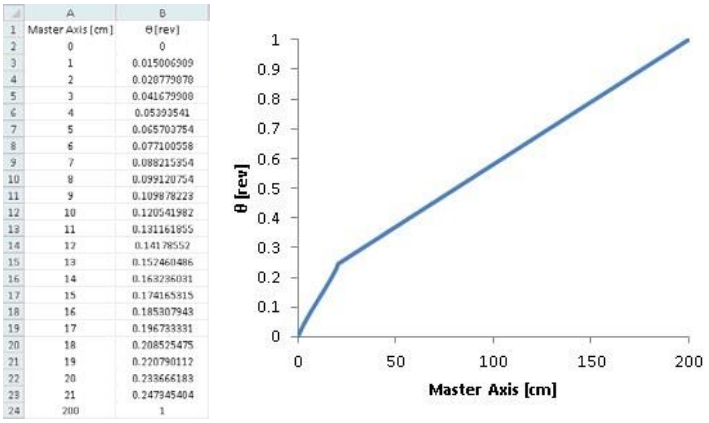


Figure 4 – Rotary Knife Linear Make Up Move

The final step is to create one Cam Table from the Cut Region and the Make-Up Move. This Cam Table is graphed in Figure 5 and is loaded into Studio 5000 to control the machine.

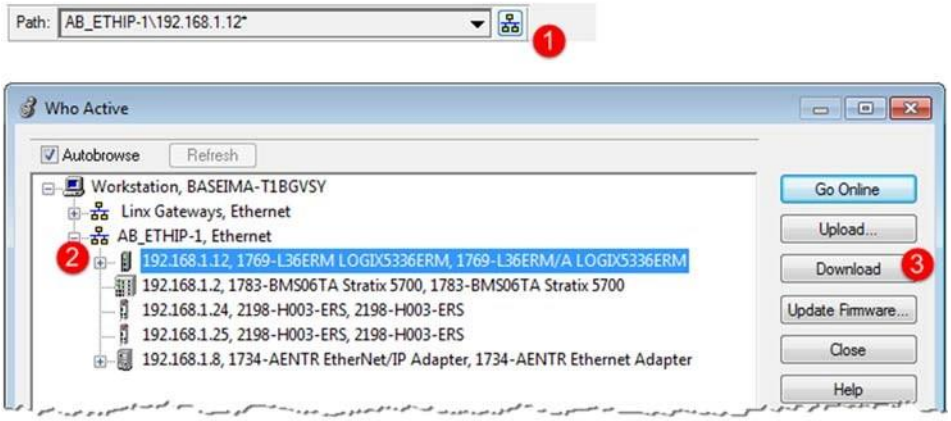


	A	B
1	Master Axis [cm]	θ [rev]
2	0	0
3	1	0.015006909
4	2	0.028779878
5	3	0.041679908
6	4	0.05393541
7	5	0.065703754
8	6	0.077100558
9	7	0.088215954
10	8	0.099120754
11	9	0.109878223
12	10	0.120541982
13	11	0.131161855
14	12	0.14178552
15	13	0.152460486
16	14	0.163236031
17	15	0.174165315
18	16	0.185307943
19	17	0.196733331
20	18	0.208525475
21	19	0.220790112
22	20	0.233666183
23	21	0.247345404
24	200	1

Figure 5 – Rotary Knife Cam Profile

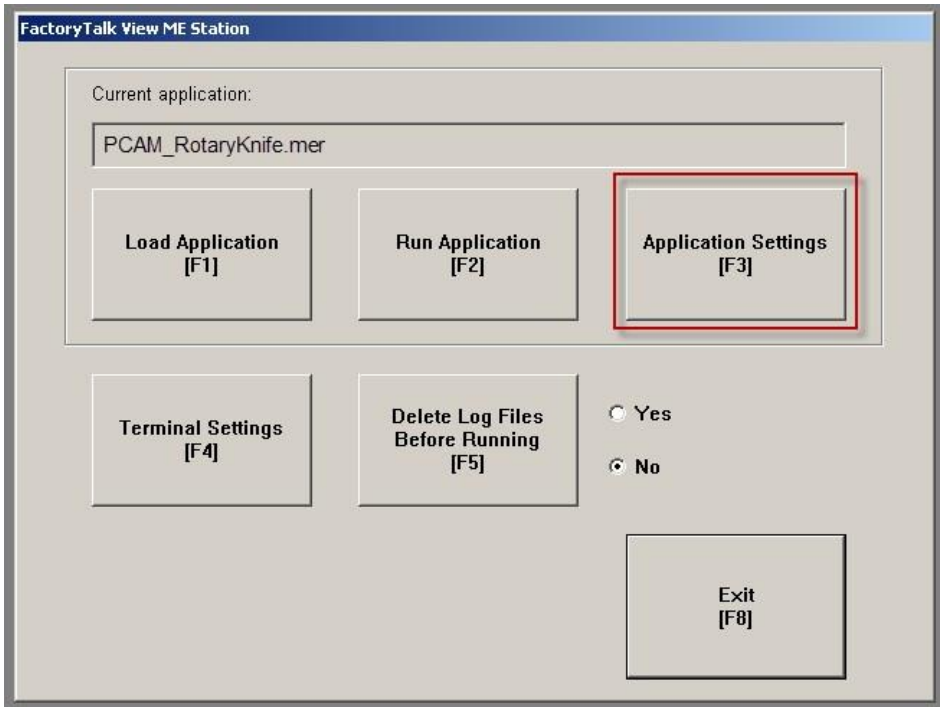
Review Code and Execute a Cam Profile

1. Open file **\Desktop\Lab Files\PCAM Rotary Knife Lab\PCAM_RotaryKnife_Begin.ACD**.
2. Download the program to the controller at **192.168.1.12**.
   1. Click **'Who Active'** button.
   2. Select controller from the Ethernet driver.
   3. Click **Download** button.
   4. On pop-up dialog window, click the **Download** button.

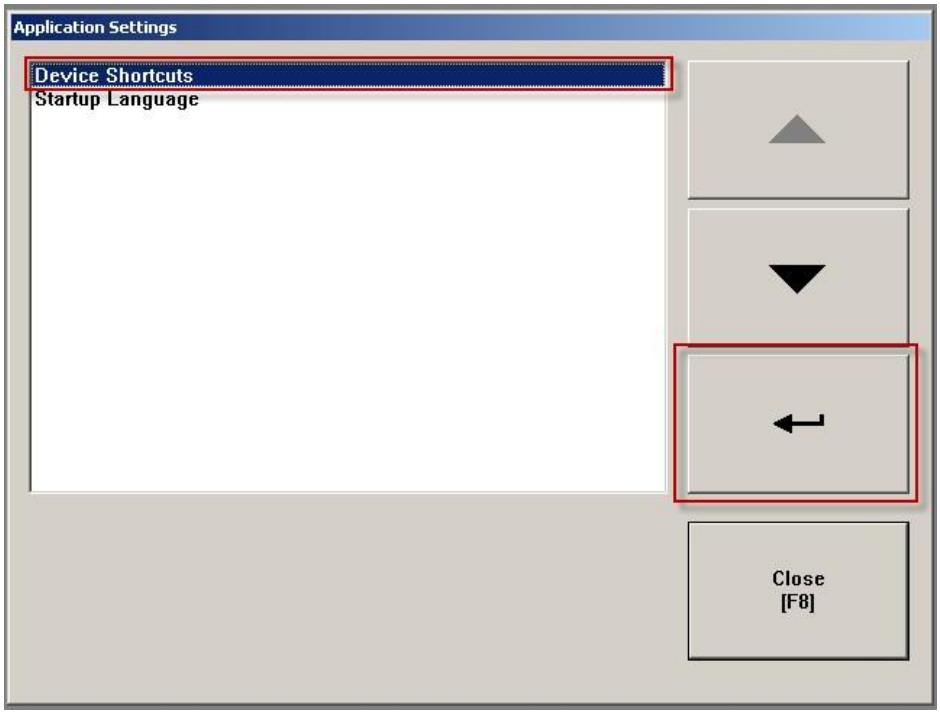


3. Once the program is downloaded, set the controller to **Run Mode**.
4. Open file **\Desktop\Lab Files\PCAM Rotary Knife Lab\PCAM_RotaryKnife.mer**, by double-clicking it, to run the lab HMI.
5. On the **'Replace Local System Directory'** dialog, press the **Yes** button to continue.

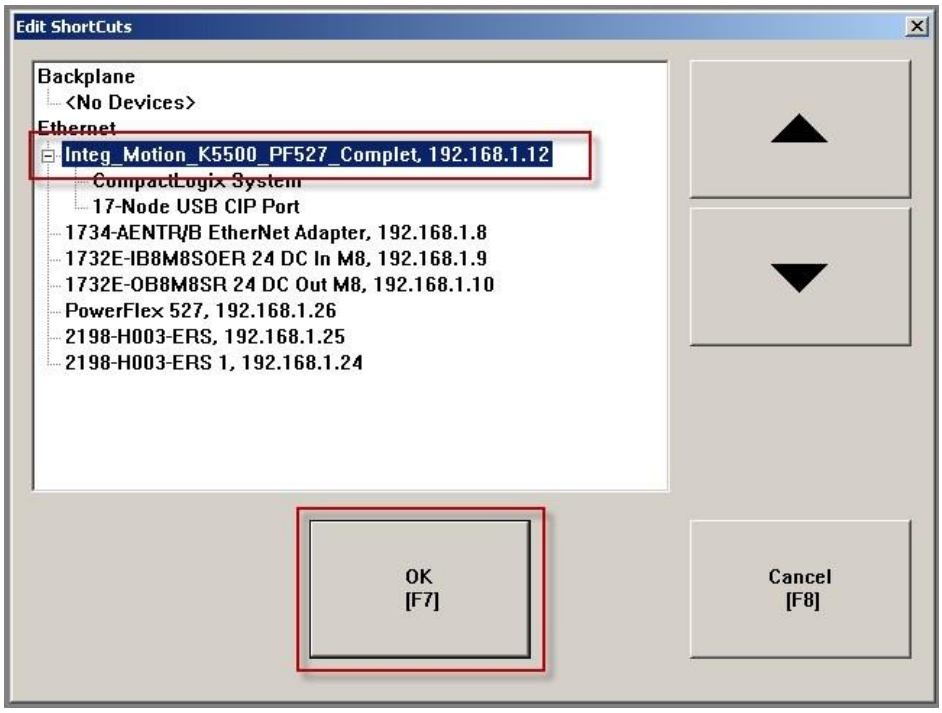
6. Click **Application Settings**



7. Select **Device Shortcuts** and hit the Enter button.



8. Verify that the controller at **192.168.1.12** IP address is selected as shown below. Click **OK**.

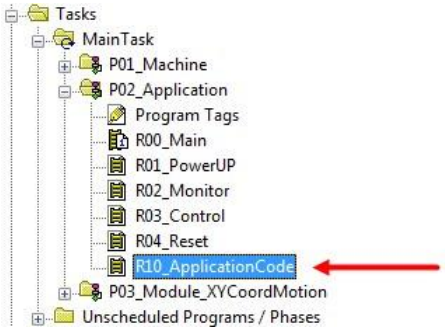


Click the **Close** button.

9. On the **'FactoryTalk View ME Station'** dialog, press the **Run Application** button.

10. On the HMI press the **'Clear Faults'** button to reset the system.

11. From the Controller Organizer navigate to the routine **Main Task > P02_Application > R10_ApplicationCode** and open it.



12. Let's review the application code.

Rung 1 - Orient Rotary Knife to the beginning of the Cut Region; a -1/8 rev move relative to home

Rung 2 - Home the Conveyor (Master Axis – AXIS_01)

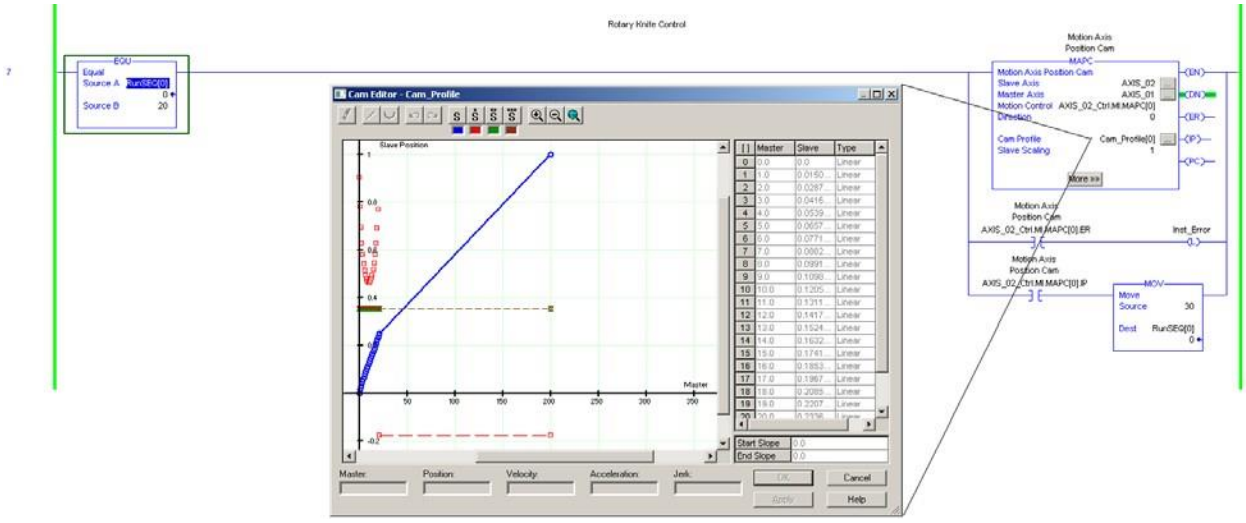
Rung 3 - Home the Rotary Knife (Slave Axis – AXIS_02)

Rung 4 - Read the Switch (POINT I/O 1734-IB8 IN0 – on controller demo) to determine whether to use the Default (0) or User Defined (1) Cam Table

Rung 5 - If IN0 == 0 then Load the Default Cam Table (Cam Profile 1)

Rung 6 - If IN0 == 1 then Load User Defined Cam Table (Cam Profile 2)

Rung 7 - Control the Rotary Knife based on the Master Axis and Cam Table (MAPC)



Rung 8 - Read the ANALOG INPUT 0 (rotary knob on the controller demo) and set the Feed Rate

Rung 9 - Start the Conveyor at the specified Feed Rate which in turn starts the Rotary Knife

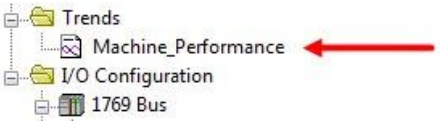
NOTE: In order to modify the Cam Table selection (IN0) or Feed Rate (ANALOG INPUT 0) the machine must be brought to a stop and then restarted.

13. The **ANALOG INPUT 0** rotary knob defines the machine Feed Rate and the range is from 0 to 200 cm/s. As a starting point, **set the rotary knob to 5.0** which equates to half speed.

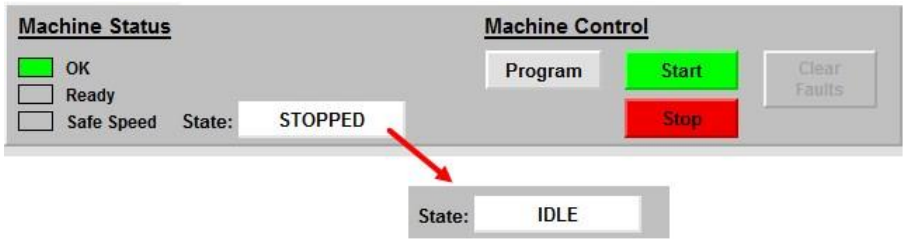
14. Make sure the **POINT I/O 1734-IB8 IN0** is set to the **left (0)** to load the default Cam Table (Cam Profile 1).

15. Manually align the white strip on the load disk of the right hand motor (Knife - AXIS_02) to face toward you. This represents the knife blade and where the product is cut.

16. We want to monitor the machine performance in a trend. Under **Trends** in the Controller Organizer, double-click on the trend **Machine_Performance** to open it.



17. Click **Run** to start collecting data in the trend.
18. On the HMI, press the **Start** button to ready the system for motion. The Machine State should transition from ‘Stopped’ to ‘Idle’.



19. Press the **Start** button again to start executing the motion application code.

    The Rotary Knife is reoriented to the beginning of the Cut Region, the axes are homed, the Conveyor is started, and the Rotary Knife begins cutting the Product.

20. Now go back to the **Trend** screen. Once the Rotary Knife has run several cycles you can stop collecting data by pressing the **Stop** button.

21. Press the **Stop** button on the HMI to stop motion.

22. Let's review the signals monitored by the trend.

    AXIS_01.ActualPosition Corresponds to the Master Axis Product [cm]. The distance between cuts should equal the Cut Length

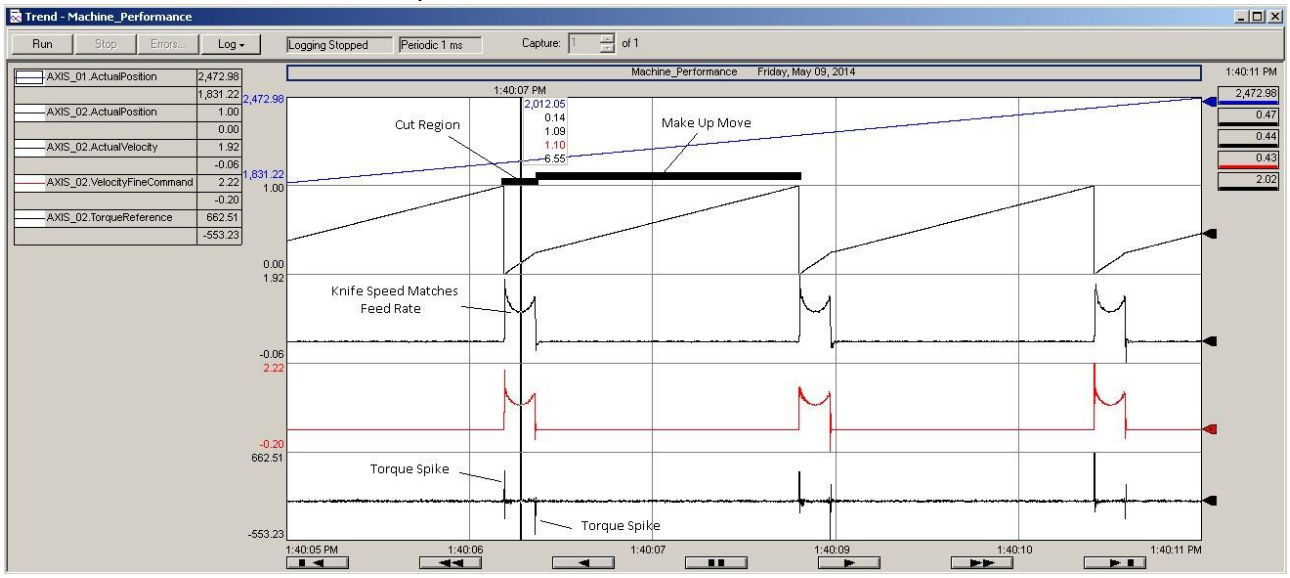
    AXIS_02.ActualPosition Corresponds to the orientation of the Rotary Knife [rev]

    AXIS_02.ActualVelocity Corresponds to the angular velocity [rev/s]

    AXIS_02.VelocityFineCommand Corresponds to the command velocity

    AXIS_02.TorqueReference Corresponds to the torque applied to the Rotary Knife [% Rated] and it is desirable to minimize the torque level

23. Now review the trend to see the performance of the machine.



There are several items to observe that include:

- Area of the Cut Region and the Make-Up Move.
- At the center of the Cut Region the Rotary Knife Velocity Command and Actual Velocity equal the machine Feed Rate.

- Compare the Knife Velocity [*cm/s*] = $2\pi \left[\frac{\quad}{\quad}\right]$ to the speed of the Conveyor by looking in the “configurable_speed_cm_per_second” variable (rung 8).
- The torque spikes at the beginning and end of the Cut Region.

24. Now let’s observe how the Feed Rate affects the performance. Return to **Step 10** and modify the Feed Rate by setting the **ANALOG INPUT 0** to a different setting such as **2.0**, and repeat the process to see the change.

Designing a Basic Cam Table

Now we will create a Cam profile using Microsoft Excel and then adding it to the ControlLogix Controller.

1. For this section of the lab you will need to take the controller Offline to edit a Cam profile. Do that now.

2. From the Start menu in Windows, run Microsoft Excel 2010 with a blank spreadsheet.

3. In cells **E1, E2 and E3** enter the configuration variables including units as; **Feed Rate [cm/s], Knife Radius [cm], and Sample Rate [s]**. Load the initial values in cells **F1, F2, F3** as; **10cm/s, 15cm, and 0.1s**.

	E	F
1	feed rate [cm/s]	10
2	knife radius [cm]	15
3	sample rate [s]	0.10

4. In cells **A1, B1 and C1** label the system dependent variables including units as; **Master Axis [cm], θ [rev], and ω [rev/s]**.

	A	B	C
1	Master Axis [cm]	θ [rev]	ω [rev/s]

5. In cells **A2, B2 and C2** load the initial values; **0, 0, and =F$1/(2*3.14*COS(2*3.14*B2-3.14/4)*F$2)**. You can use copy/paste for the formula.

   This is the Rotary Knife inverse kinematics equations. Cell C2 should show the calculated value of 0.150063088.

6. Now copy the following equations into the following cells; **A3 =Master Axis =A2+F$1*F$3, B3 = θ =B2+C2*F$3 and C3 = ω =F$1/(2*3.14*COS(2*3.14*B3-3.14/4)*F$2)**. Make sure to copy the preceding equal sign as part of the formula.

7. Highlight **row 3** as shown and place the cursor in the bottom right corner of the selected cells to drag & fill downward until the angle reaches approximately **0.25 (row 23)**.

	A	B	C
1	Master Axis [cm]	θ [rev]	ω [rev/s]
2	0	0	0.150069088
3	1	0.015006909	0.137729691

   This completes the Cam Table for the Cut Region.

	A	B	C
1	Master Axis [cm]	θ [rev]	ω [rev/s]
2	0	0	0.150069088
3	1	0.015006909	0.137729691
4	2	0.028779878	0.129000304
5	3	0.041679908	0.122555016
6	4	0.05393541	0.117683439
7	5	0.065703754	0.113968037
8	6	0.077100558	0.111147962
9	7	0.088215354	0.109054002
10	8	0.099120754	0.107574689
11	9	0.109878223	0.106637596
12	10	0.120541982	0.106198729
13	11	0.131161855	0.106236643
14	12	0.14178552	0.10674966
15	13	0.152460486	0.10775545
16	14	0.163236031	0.109292844
17	15	0.174165315	0.111426276
18	16	0.185307943	0.114253883
19	17	0.196733331	0.117921439
20	18	0.208525475	0.122646376
21	19	0.220790112	0.128760708
22	20	0.233666183	0.13679221
23	21	0.247345404	0.147630547

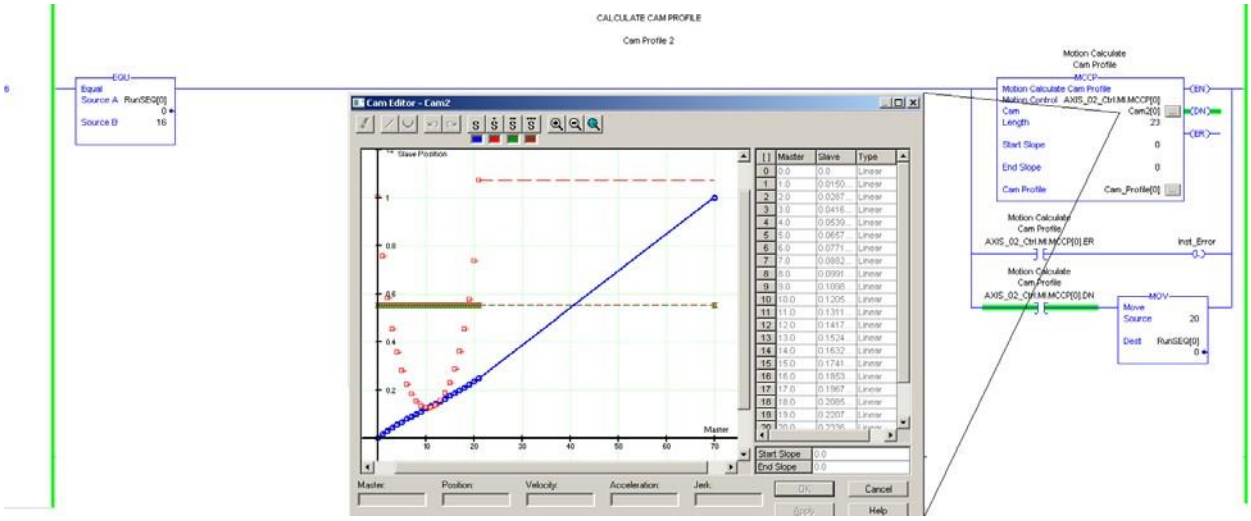
8. Now we need to update the Cam Table to include the Make-Up Move. This time the Product Cut Length will be 70cm. Thus the Master Axis goes to **70**; enter this value into cell **A24**. The Rotary Knife rotates to **1**; enter this value into cell **B24**.

	A	B
24	70	1

9. Copy columns **A and B** from row **2 to 24** as shown.

	A	B
1	Master Axis [cm]	θ [rev]
2	0	0
3	1	0.015006909
4	2	0.028779878
5	3	0.041679908
6	4	0.05393541
7	5	0.065703754
8	6	0.077100558
9	7	0.088215354
10	8	0.099120754
11	9	0.109878223
12	10	0.120541982
13	11	0.131161855
14	12	0.14178552
15	13	0.152460486
16	14	0.163236031
17	15	0.174165315
18	16	0.185307943
19	17	0.196733331
20	18	0.208525475
21	19	0.220790112
22	20	0.233666183
23	21	0.247345404
24	70	1

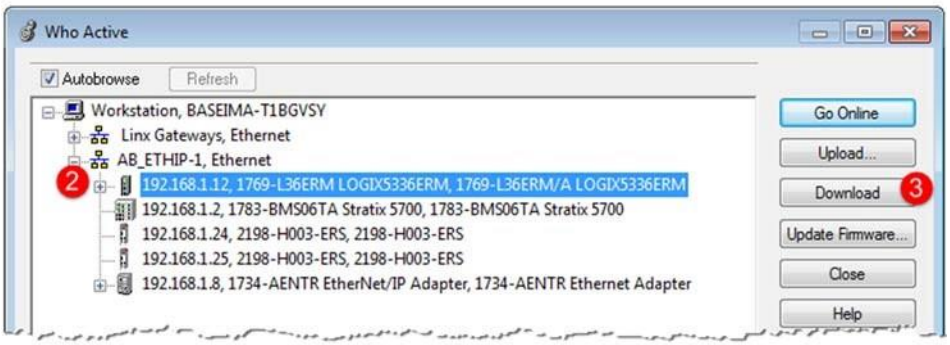
10. Highlight any data in the **Cam2** input of the MCCP command on **rung 6** and **Delete** any data. Now paste the Excel data into **Cam2** input, loading the Cam Table into the controller.



At this point you will configure the machine to operate by specifying the Operating Points.

11. Prior to executing the User-Defined Cam ponder the following questions:
    - How does the Feed Rate differ compared to the original results?
    - How does the Product Cut Length compare to the original design?
    - What difference do you expect to see in the Torque Reference?

12. Download the program to the controller at **192.168.1.12**.
    1. Click **'Who Active'** button.
    2. Select controller from the Ethernet driver.
    3. Click **Download** button.
    4. On pop-up dialog window, click the **Download** button.



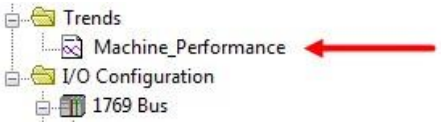
13. Once the program is downloaded, set the controller to **Run Mode**.

14. The **ANALOG INPUT 0** rotary knob defines the machine Feed Rate and the range is from 0 to 200 cm/s. As a starting point, **set the rotary knob to 5.0** which equates to half speed.

15. Make sure the **POINT I/O 1734-IB8** input **IN0** is set to the **right (1)** to load the User-Defined Cam Table (Cam Profile 2).

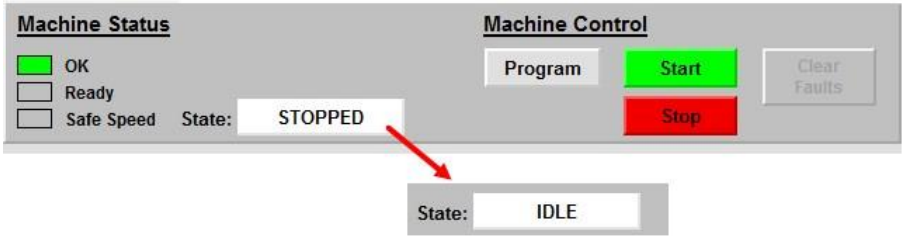
16. Manually align the white strip on the load disk of the right hand motor (Knife - AXIS_02) to face toward you. This represents the knife blade and where the product is cut.

17. We want to monitor the machine performance in a trend. Under **Trends** in the Controller Organizer, double-click on the trend **Machine_Performance** to open it.



18. Click **Run** to start collecting data in the trend.

19. On the HMI, press the **Start** button to ready the system for motion. The Machine State should transition from 'Stopped' to 'Idle'.



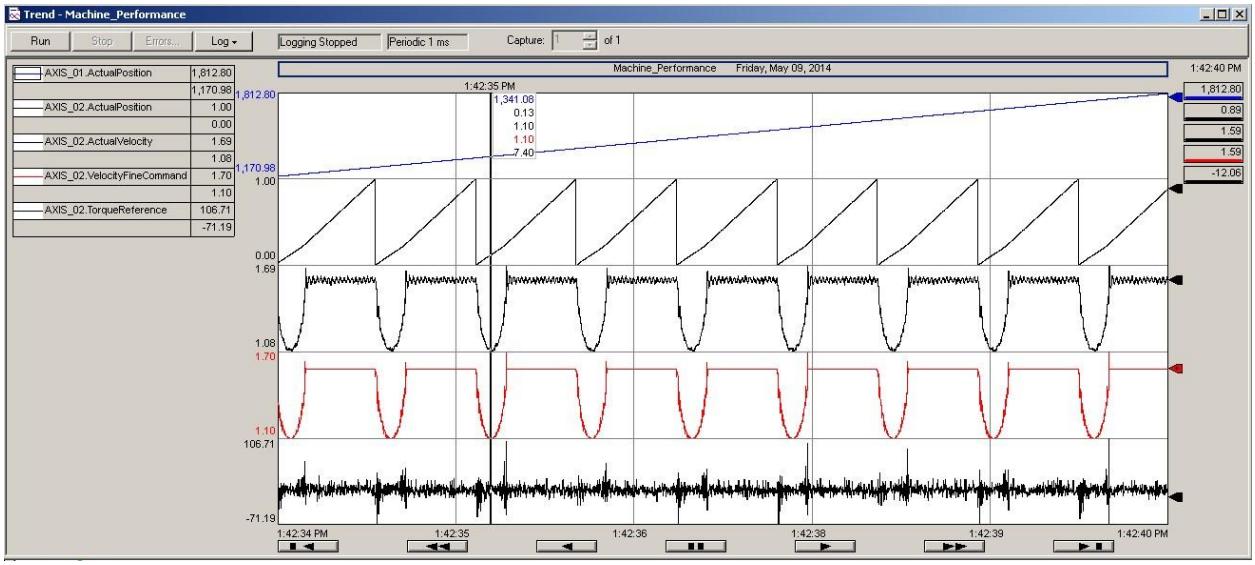
20. Press the **Start** button again to start executing the motion application code.

    The Rotary Knife is reoriented to the beginning of the Cut Region, the axes are homed, the Conveyor is started, and the Rotary Knife begins cutting the Product.

21. Now go back to the **Trend** screen. Once the Rotary Knife has run several cycles you can stop collecting data by pressing the **Stop** button.

22. Press the **Stop** button on the HMI to stop motion.

23. The machine performance is shown below.

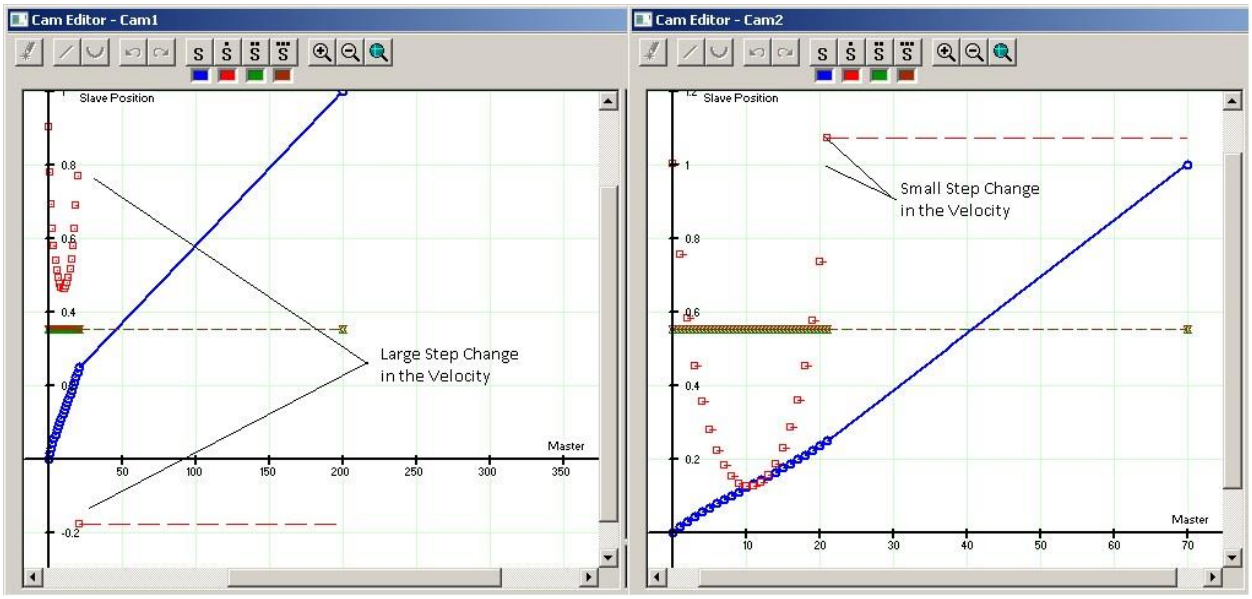


There are several items to observer that includes:

- Area of the Cut Region and the Make-Up Move.
- At the center of the Cut Region the Rotary Knife Velocity Command & Actual Velocity equals the machine Feed Rate.
- Compare the Knife Velocity[*cm/s*] = 2π $\left[\frac{\quad}{\quad}\right]$ to the speed of the Conveyor by looking in the "configurable_speed_cm_per_second" variable (rung 8).
- The previous torque spikes at the beginning and end of the Cut Region are not present.

24. Why does the User-Defined Cam Table, that has a shorter Cut Length and produces more Products per second, require less torque?

    This example illustrates the importance of the Cam Design. In this specially chosen example, a Cut Length was chosen that allows the boundary conditions between the Cut Region and the Make-Up Move to match. The graph shown below shows the Default Cam Table (Cam1) and the User-Defined Cam Table (Cam2). By observing the Cam Profiles velocity command, in red, the designer can observe the severe step response in the velocity command. This step response is what causes the large torque spikes.



If it is necessary to achieve this level of performance for arbitrary Cut Lengths, then Advanced Cam Designs are necessary that match the boundary conditions.

This concludes this lab.

Appendix D: Tuning Techniques Lab

In this lab a tuning procedure is presented that provides a solution that compensates for unknown mechanics, produces high performance results, and does not require expertise in tuning. It has been observed that this procedure produces satisfactory results for nearly 95% of the applications our customers encounter.

There are two parts to the procedure. The first part leverages the load observer which is a feature internal to the drive. The load observer estimates the torque required to move the mechanical load in real time and adds the result to the torque command.
This automatically compensates for a wide range of unknown mechanics.

The second part compensates for problematic mechanics which can arbitrarily limit the performance. This mechanical limitation is often referred to as a mechanical resonance. By identifying the resonance, the Torque Notch Filter can be configured to compensate for the limitation. Currently the identification and setting of the Torque Notch Filter is a manual operation. In the future, this manual operation will automatically be configured in servo and standard drives that support the Adaptive Tuning Features.

In this lab you will be shown how to appropriately configure the Load Observer, and the simple process for identifying and configuring the Torque Notch Filter Frequency to easily tune most mechanical systems. The topic of Advanced Tuning Techniques which covers the remaining 5% of applications is not covered as part of this lab.

Lab Content Includes

- Design Procedure: Describes how the key features function
- Step-by-Step Procedure: Configuring the Default Configuration

- Appendix: Step-by-Step Procedure - Identifying and Compensating for a Mechanical Resonance

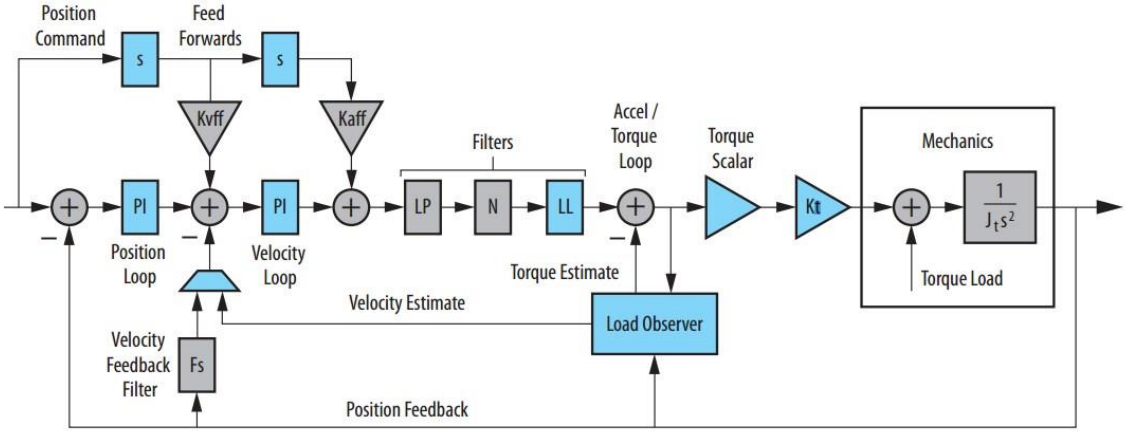
Key Topics Covered

- Tuning Strategy for 5x00 & 6500 Kinetix servo drives
- Tuning-less Configuration & Possible Modifications

Design Procedure: Describe How the Key Features Function

In this section, a description of the features that are leveraged as part of the tuning procedure are presented. A complete understanding of these features are not necessary, to execute the tuning procedure, but are included for those who have experience designing control algorithms. At any point, due to time constraints, if you struggle to grasp the design, continue to progress through the lab, or consult the lab instructor for more details.

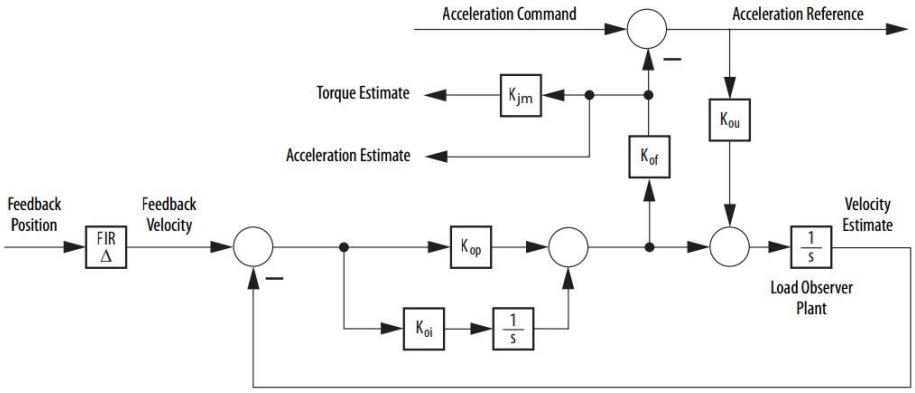
The Kinetix servo drives implement an acceleration/torque loop, which is nested within a velocity PI control loop, which is nested within an outer position PI control loop are illustrated below.



The multi-loop PI control structure is common among servo drives. There are slight differences between drive families based on the implementation and specific units which are not covered as part of this lab. This lab will focus on the details pertaining to the Load Observer and Torque Notch Filter (N).

Load Observer: The load observer feature is part of the control loop inside the drive that estimates the mechanical load on the motor and compensates for it, thereby forcing the motor to behave as if it is unloaded and relatively easy to control. As a result, the load observer automatically compensates for disturbances and load dynamics, such as inertia/torque changes, compliance, backlash, and resonances that are within the velocity loop bandwidth.

The load observer acts on the acceleration signal within the control loops and monitors the Acceleration Reference and the Actual Position feedback. The load observer models an ideal unloaded motor and generates a load Torque Estimate, in torque units, that represents any deviation in response of the actual motor and mechanics from the ideal model. This deviation, shown below, represents the reaction torque placed on the motor shaft by the load mechanics. It is estimated in real time and compensated by closed loop operation.



The load observer also generates a Velocity Estimate signal that you can apply to the velocity loop. The Velocity Estimate has less delay than the Velocity Feedback signal derived from the actual feedback device. It also helps to reduce high frequency output noise caused by load observer's aggressive action on the acceleration reference. Together, the Load Observer with Velocity Estimate provides the best overall performance when the Axis Configuration is set to Position Loop. The Position Loop configuration incorporates most applications.

The Load Observer has two configurable parameters which are the Load Observer Bandwidth and the Load Observer Integrator Bandwidth. For most applications and the purposes of this lab the Load Observer Integrator Bandwidth will remain 0. The primary configuration is the Load Observer Bandwidth. It can be tuned similarly to the Position and Velocity Loop Bandwidths. For the tuning procedure presented in this document a satisfactory configuration is to set it equal to 1 / (2*π*Drive Model Time Constant). The Drive Model Time Constant is an internal drive parameter associated with the type of drive, motor, and feedback device.

Torque Notch Filter: is a filter that passes most frequency signals unaltered but attenuates signals within a specific range of frequencies. Notch filters typically have a relatively narrow and deep attenuation band. Maximum attenuation is achieved at the notch filter frequency. The notch filter in Kinetix 5500, 6000, 6200, and 6500 drives is second order with an attenuation of approximately 40 dB at the notch frequency. The notch filter is effective in resonance control when the resonant frequency is higher than the control loop bandwidth. The notch filter works by significantly reducing the amount of energy in the drive output that can excite high frequency resonances. It can be used even when resonant frequencies are relatively close to the control loop bandwidth. That is because the phase lag introduced by the notch filter is localized around the notch frequency. For the notch filter to be effective, the Notch Filter Frequency must be set close to the mechanical resonant frequency. As a general rule of thumb the Notch Filter Frequency should not be less than 300Hz.

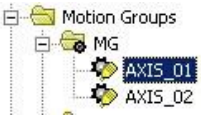
Configuring the Default Tuning Configuration

In this portion of the lab, the axis will be configured to the Default Tuning Configuration which will work without modification for approximately 80% of the applications. Depending on the version of Studio 5000, this may be the configuration when an axis is created. The general rule of thumb constrains the Velocity Loop Bandwidth = Load Observer Bandwidth / 4, Position Loop Bandwidth = Velocity Loop Bandwidth / 4, Load Observer Integrator Bandwidth = Velocity Loop Integrator Bandwidth = Position Loop Integrator Bandwidth = 0, Torque Low Pass Filter Bandwidth = 5*Load Observer Bandwidth, Load Ratio = 0, and the Load Observer Bandwidth = 1/(2*π*Drive Model Time Constant).

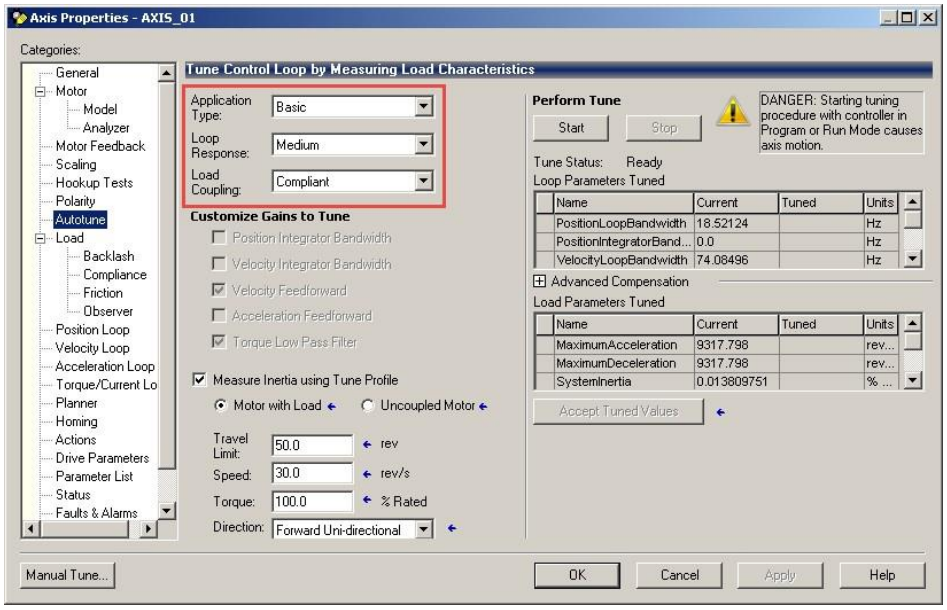
In the included ACD file Axis_01 has appropriately been configured. Steps 2-6 are shown to illustrate the appropriate setup procedure in the event the axis was not configured. After understanding the desired

configuration you will be asked to appropriately configure Axis_02. Finally, the program is executed to illustrate the performance of the tuning strategy.

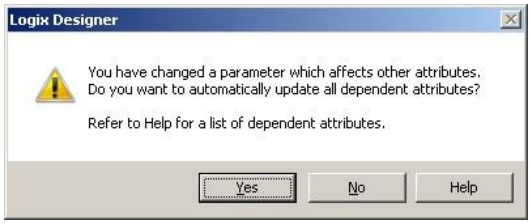
1. Open file **\Desktop\Lab Files\Tuning Techniques Lab\Tuning_Begin.ACD**.
2. Right-click on **Axis_01** and select **Properties**.



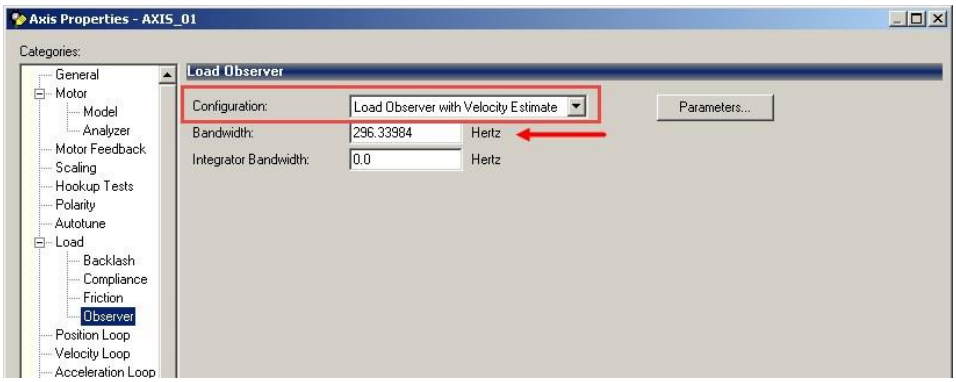
3. Choose the **Autotune** category and verify the Application Type is set to Basic, Loop Response to Medium and Load Coupling to Compliant. Make any necessary changes and click **Apply**.



Note: If the following window appears click “yes” unless you are specifically modifying one of the dependent attributes and do not want to reconfigure all of the tuning parameters. Always reconfirm the tuning parameters if this box appears.

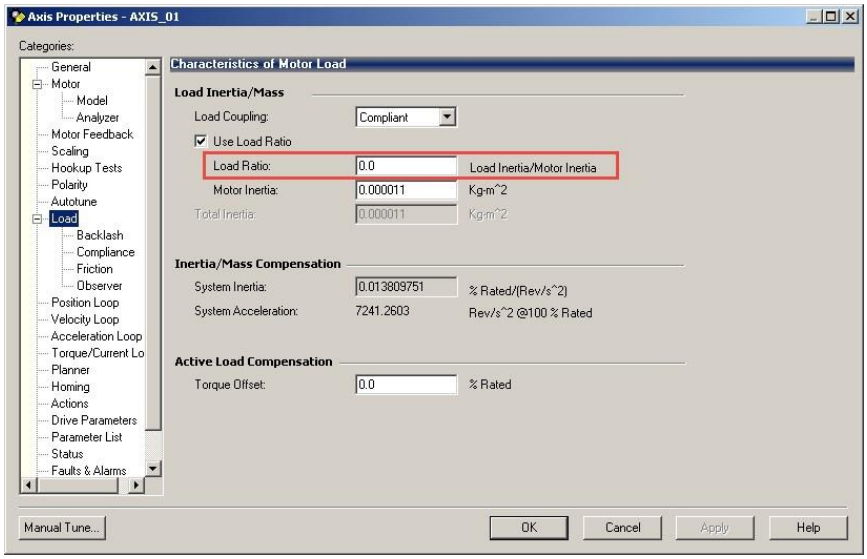

Logix Designer
You have changed a parameter which affects other attributes.
Do you want to automatically update all dependent attributes?
Refer to Help for a list of dependent attributes.
Yes
No
Help

4. Choose the **Load -> Observer** category and verify the Load Observer Configuration is set to Load Observer with Velocity Estimate. Make any necessary changes and click **Apply**.



Make note of the Load Observer Bandwidth value.

5. Choose the **Load** category verify the Load Ratio is set to 0. Make any necessary changes and click **Apply**.



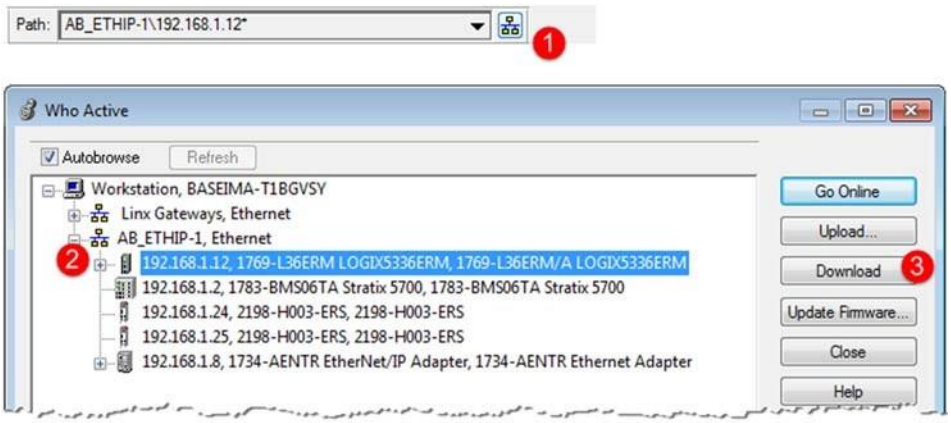
6. Confirm the appropriate tuning parameters are loaded in the Axis Properties …
   - Click **'Load - Compliance'** and confirm the Low Pass Filter Frequency = 5 * Load Observer Bandwidth.
   - Click **'Velocity Loop'** and confirm the Velocity Loop Bandwidth = Load Observer Bandwidth / 4.
   - Click **'Position Loop'** and confirm the Position Loop Bandwidth = Velocity Loop Bandwidth / 4.

7. Now follow steps 2-6 again to configure **Axis_02** for the default configuration.

8. At this point Axis_01 and Axis_02 should be configured with the Default Parameters which provides the necessary performance for most applications.

9. Download the program to the controller at **192.168.1.12**.
   1. Click **'Who Active'** button.
   2. Select controller from the Ethernet driver.
   3. Click **Download** button.

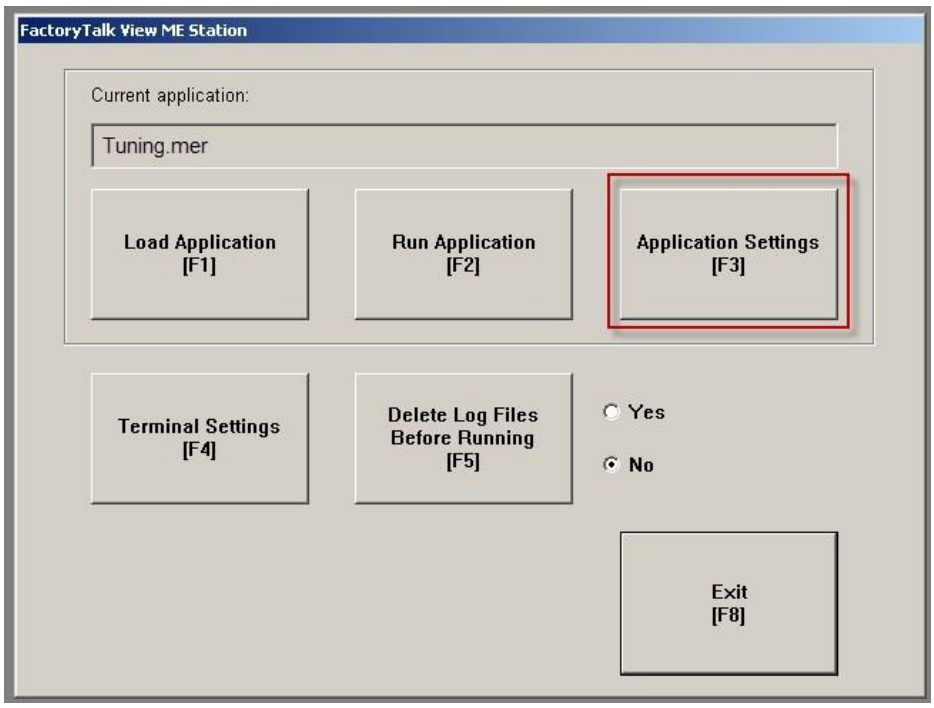
4. On pop-up dialog window, click the **Download** button.



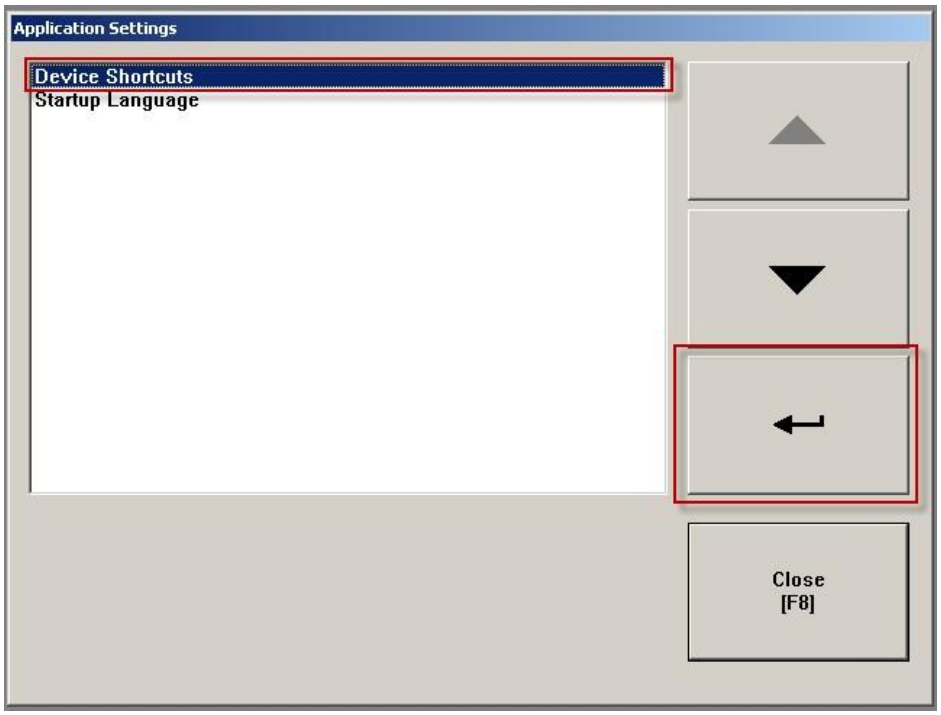
10. Set the controller to **Run Mode**.
11. Open file **\Desktop\Lab Files\Tuning Techniques Lab\Tuning.mer**, by double-clicking it, to run the lab HMI.

12. On the **'Replace Local System Directory'** dialog, press the **Yes** button to continue. 13.

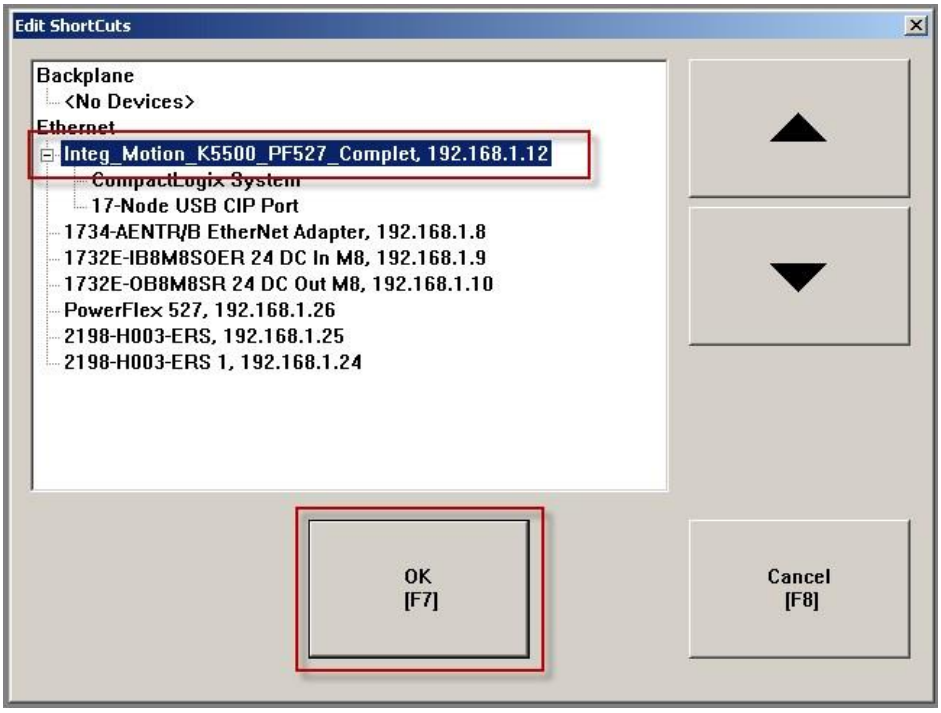
    Click **Application Settings**



14. Select **Device Shortcuts** and hit the Enter button.



15. Verify that the controller at **192.168.1.12** IP address is selected as shown below. Click **OK**.

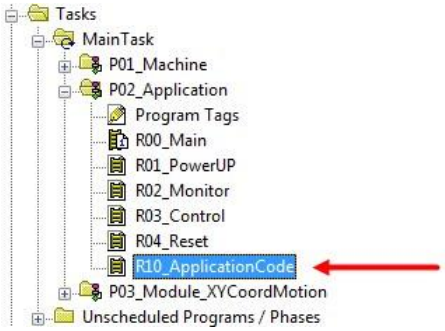


Click the **Close** button.

16. On the **'FactoryTalk View ME Station'** dialog, press the **Run Application** button.

17. On the HMI press the **'Clear Faults'** button to reset the system.

18. From the Controller Organizer navigate to the routine Main Task > P02_Application > R10_ApplicationCode and open it.



19. Visually review the Application Code Sequencing: Upon starting the application the program will

    Rung 1 – Home Axis_01

    Rung 2 – Home Axis_02

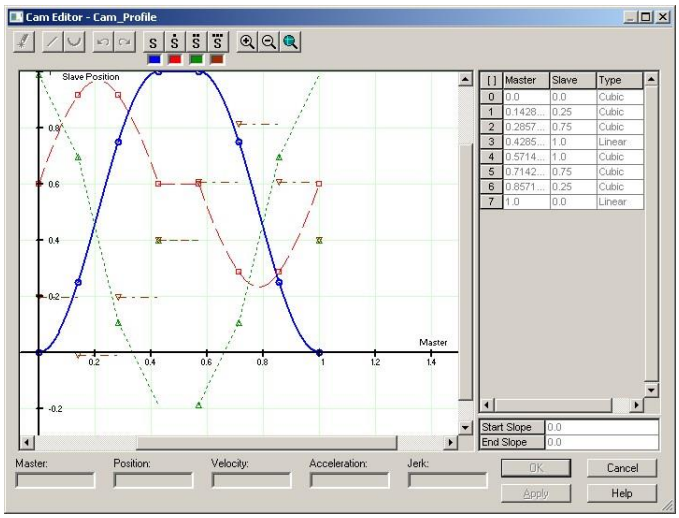
    Rung 3 – Illustrate how

    the default Load

    Observer Bandwidth is

    determined

    Rung 4 – Run Axis_01 with a smooth indexing profile to minimize backlash on the axis

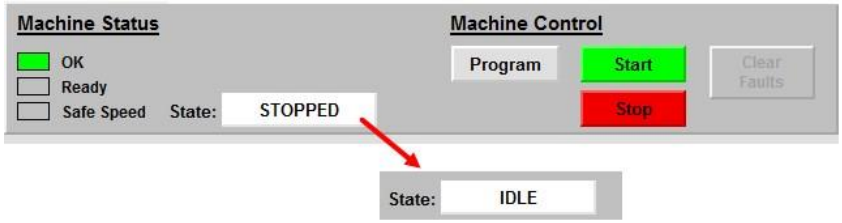


    Rung 5 – Run Axis_02 the same as Axis_01

20. We want to monitor the machine performance in a trend. Under Trends in the Controller Organizer, double-click on the trend Results to open it.

21. Click **Run** to start collecting data in the trend.

22. On the HMI, press the **Start** button to ready the system for motion. The Machine State should transition from ‘Stopped’ to ‘Idle’ state.



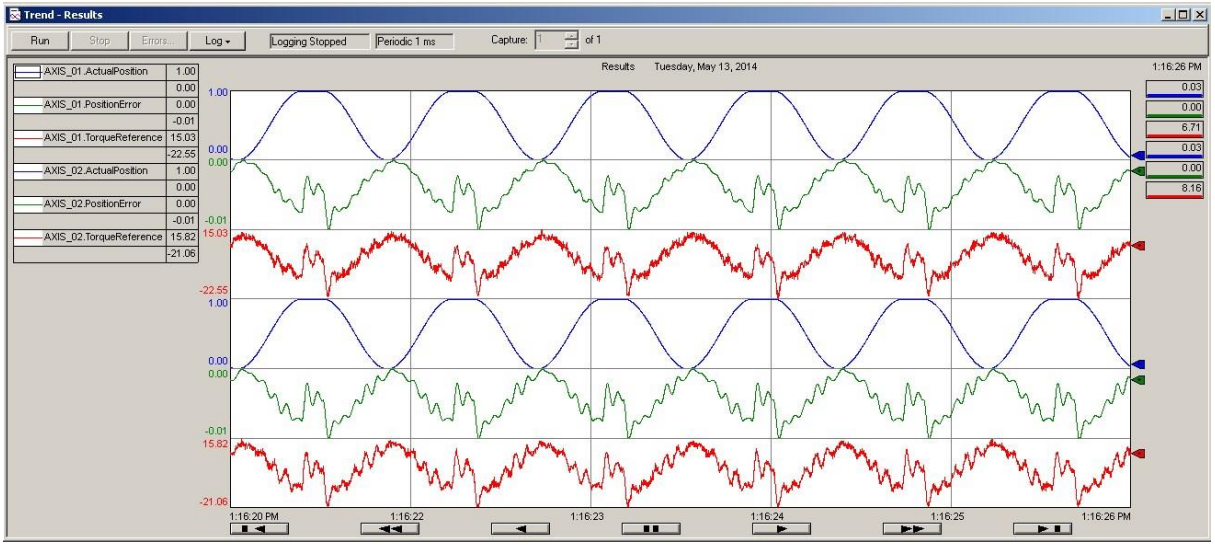
23. Press the **Start** button again to start executing the motion application code.

    Each axis should start indexing back and forth in a similar motion.

24. Now go back to the trend screen. Once the axes have ran several cycles press the **Stop** button to stop collecting data.

25. Press the **Stop** button on the HMI to stop motion.

26. Visually review the performance of the axes in the trend.



Notice that even with the large inertia, i.e. a load ratio of approximately 32:1, and significant lost motion on Axis_01 the default tuning settings in conjunction with the smooth Cam Profile nicely controls the axes.

Identify and Compensate for Mechanical Resonances

For the large inertia loads, used in this lab, a mechanical resonance does not cause a significant problem thus the Torque Notch Filter does not need to be configured. In this appendix, a general description of mechanical resonance is presented along with a step by step procedure to identify and compensate for the mechanical resonances.

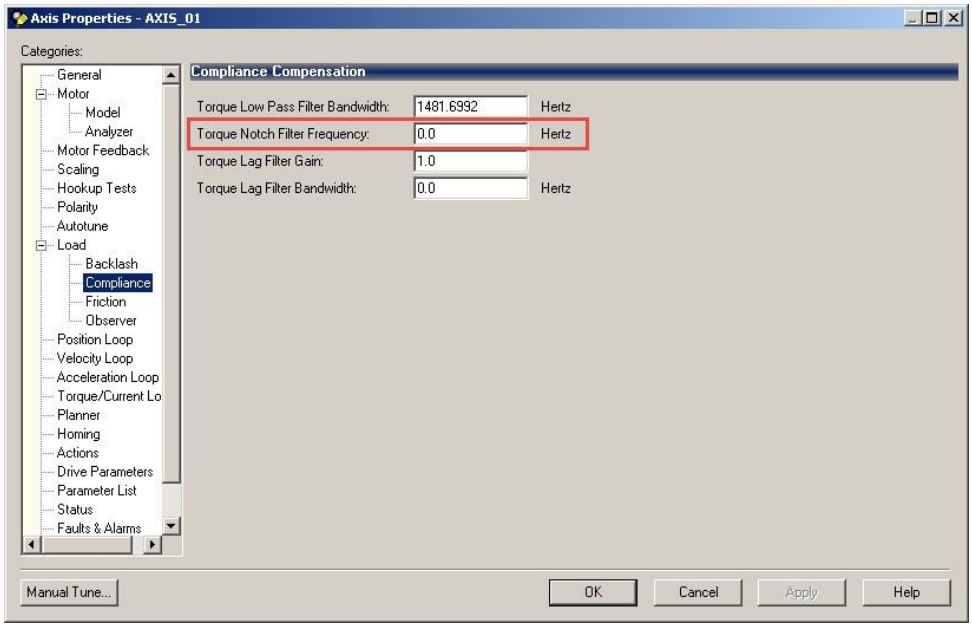
What are common causes of mechanical resonances? Typical causes can include coupling, encoders, and compliant system. What are typical symptoms? They include loud pitch noises that are typically a single frequency above 300Hz. Why is it important to minimize the mechanical resonances? Mechanical resonances are problematic because they cause excessive ware, waste power, and reduce the performance. How do you compensate for a mechanical resonance? The two primary methods of compensating for a mechanical resonance include a costly redesign of the mechanics or a quick software modification that programmatically compensates for the resonance. This procedure focuses on the programmatic approach to compensate for the mechanical resonance by setting the Torque Notch Filter. For complex mechanics it may also be necessary to decrease the Torque Low Pass Filter or decrease the servo loop gains.

1. Perform the following move sequence, by using Motion Direct Commands, to excite the resonance.
    1. Enable the drive with an MSO instruction.
    2. Slowly jog the axis with a MAJ or MAM instruction.
    3. Stop the axis with a MAS instruction.
    4. Disable the drive with an MSF instruction.

2. Determine if an audible high-frequency resonance exists in your motion application.
    - If an audible high frequency resonance is not present during the move sequence, skip the remaining steps and tuning is complete.
    - If an audible high frequency resonance is present during the move sequence, use a FFT (Fast Fourier Transform) smart phone or tablet application to identify the dominant resonant frequencies. When the Adaptive Tuning Features become available there will be features internal to the drive to identify mechanical resonances.

3. If a resonance is below the 1/(2πDrive Model Time Constant) Hz, i.e. Load Observer Bandwidth, and a low pitch growling sound is present, then an instability is present and the servo loop gains must be decreased before continuing with the following steps.

4. If a resonance is above the 1/(2πDrive Model Time Constant) Hz and a high pitch sound is present, then a mechanical resonance is present and you must set the Torque Notch Filter Frequency to the identified audible frequency.



In 95% of the application utilizing the default tuning gains and setting the Torque Notch Filter Frequency will provide adequate performance. In the remainder of the applications it may necessary to compensate for more complicated mechanics, multiple resonant frequencies, or reduce the tracking error. These situations are beyond the scope of this lab.

You may discuss these situations with the lab instructor or consult you technical specialist as the situation arises.

This concludes this lab.

Appendix E: Network (CIP) Safety

The goal of this section is to provide an overview of Networked Safety for servo drives, enabled by CIP Safety. During this section, you will:

- Learn about the basic safety standards applied to servo drives and variable frequency drives.
- Examine the difference between various methods for Safe Torque Off (STO).
- See how to configure a Kinetix 5500 drive with Networked Safety.
- Review and write ladder logic that could be used to execute STO in a Kinetix 5500 drive.

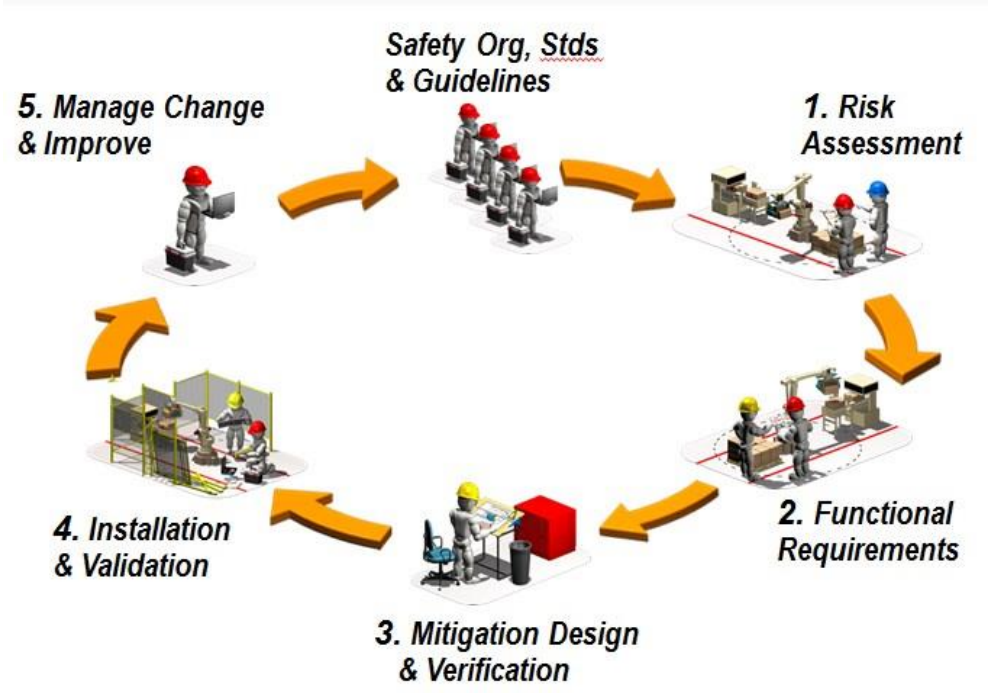
Safety Basics

Variable frequency drives, servo drives, and motors in general are covered by a variety of safety standards. These standards fit into legal frameworks in different ways, depending on the region. Some of the standards are written around components (such as a drive), and others are written around the entire machine. The drives made by Rockwell Automation that support Functional Safety are all certified by an independent third party (TÜV Rheinland) to the following product standards:

Standard	Title	Description	Kinetix 5500 Hardwired	Kinetix 5500 Networked
ISO 13849-1	Safety of Machinery - Safetyrelated Parts of Control Systems Part 1: General principles for design	Uses Performance Levels to define the risk of random dangerous failure for simple devices, including electromechanical components, and machine systems.	PLd	PLe
IEC 60261	Safety of Machinery - Functional safety of safetyrelated electrical, electronic, and programmable electronic control systems	Uses Safety Integrity Levels to define the risk of random dangerous failure for complex electronic devices, such as Programmable Automation Controllers, and machine systems.	SILCL 2	SILCL 3
IEC 61800-5-2	Adjustable speed electrical power drive systems Part 5-2: Safety Requirements - Functional	Defines the expected behavior for various safety functions that can be performed by variable frequency drives and servo drives.	Check	Check
IEC 61508	Functional safety of electrical/electronic/programmable electronic safety-related systems	Uses Safety Integrity Levels to define the risk of random dangerous failure for any scale of electronic control system, from small machines to very complex processes.	SILCL 2	SILCL 3

Certification to these standards implies that the drive can be used as a subsystem in a safety function **up to** the limit shown in the table. These certifications alone do not guarantee that the drive is implemented in the proper way. There are many aspects of the Machine Safety Lifecycle that are not covered in this tutorial that influence the overall Performance Level or Safety Integrity Level of a machine, including the:

- Risk Assessment
- Functional Requirements
- Mitigation Design & Verification
- Installation & Validation
- Change Management & Improvements

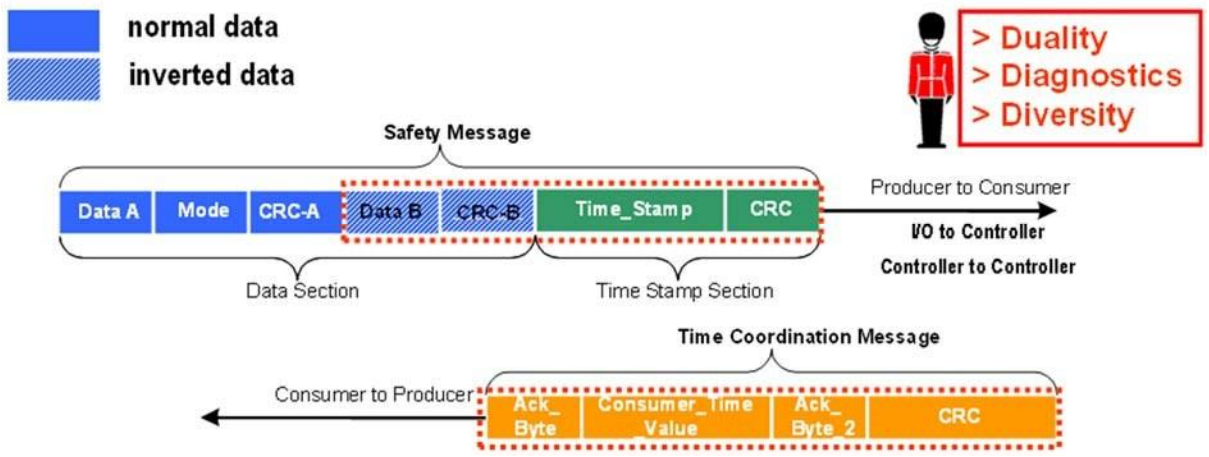


For more information on any of these areas, please visit another session during this event focused on Safety Lifecycle Management, or consult with your local Rockwell Automation or distributor resources.

Safe Torque Off (STO)

One of the most visible and common hazards on machines comes from moving parts. Since many of these parts are moving because of motors attached to them, let's focus on ways to make those motors safe. At the most basic level, there is only one safety control function that can be performed with a motor - removal of torque producing power. This was done traditionally with Lock Out Tag Out (LOTO), to remove all sources of power from a machine. More recently, control power has been left on and motor power was removed through a variety of means.

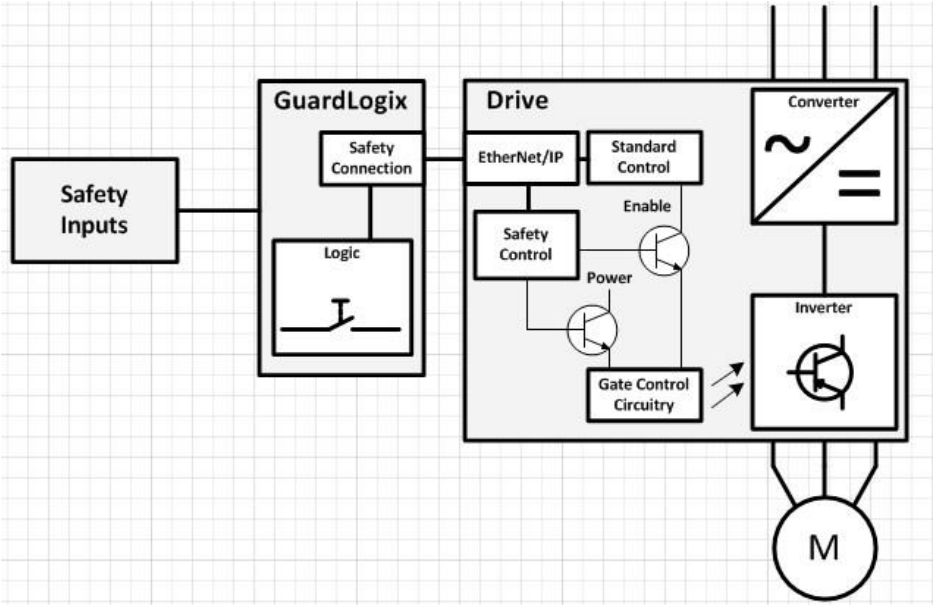
Over the last decade, communication channels between field devices and controllers have evolved to include "safe connections". The protocol used by Rockwell Automation is based on the CIP Safety standard from ODVA. This standard is designed and certified for transport of data with high integrity. This design includes sending the data over standard networks, in specialized packets to remove the chances for data corruption. This is accomplished by using basic safety principles, including Duality, Diversity, and Diagnostics.



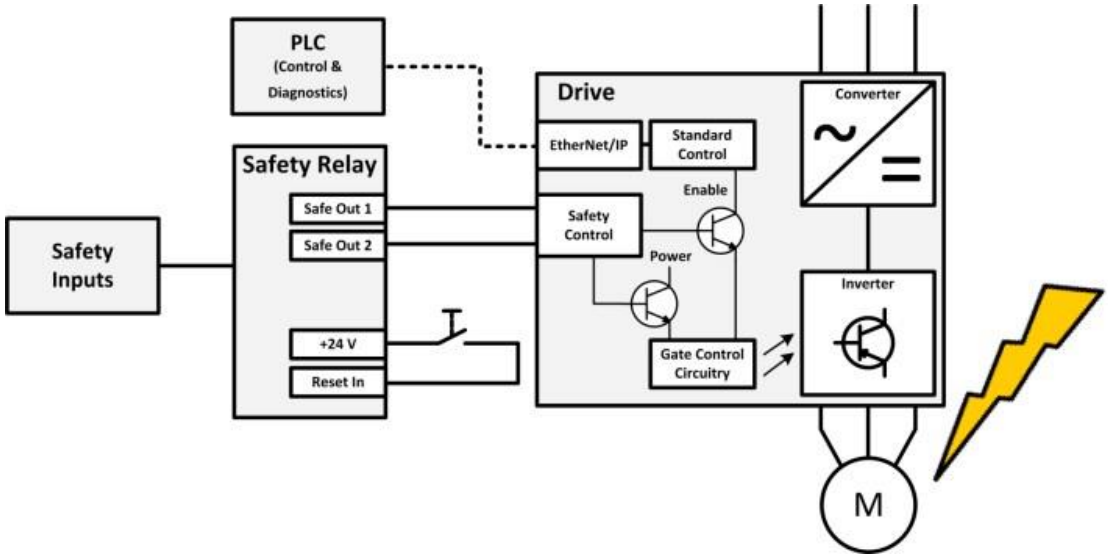
Seamless communication in the past was nearly impossible because no single network was able to integrate safety and standard control systems while also enabling the seamless transport of data across multiple plant-floor physical networks. That changed with the Common Industrial Protocol (CIP), an application protocol for industrial networking that is independent of the physical network. The CIP protocol provides a set of common services for control, configuration, collection and sharing across all of the CIP networks, DeviceNet, ControlNet and EtherNet/IP.

CIP Safety also helps eliminate the need to install expensive and difficult-to-maintain gateways between each network. Before the development of safety networks, engineers often had to use smaller systems or minimize their performance requirements since it was difficult to hard-wire interlocks and relay-based safety logic into a complete automation system. Now, engineers can integrate their devices on common physical network segments and allow safety and standard information to flow between devices and controllers.

The latest generation of Safe Torque Off drives includes the ability to safely remove torque using the network connection, with CIP Safety over EtherNet/IP. That network connection can provide tremendous diagnostics on the same wires that provide the standard control, and reduces your wiring to an absolute minimum.



Safe Torque Off should be used for routine, repetitive, predictable actions, such as clearing a jam or changing tooling. Safe Torque Off is not suitable for electrical work of any kind. While it removes the ability to create torque, there can still be hazardous voltages present on the motor terminals. This is why LOTO is still a crucial part of a safety strategy.

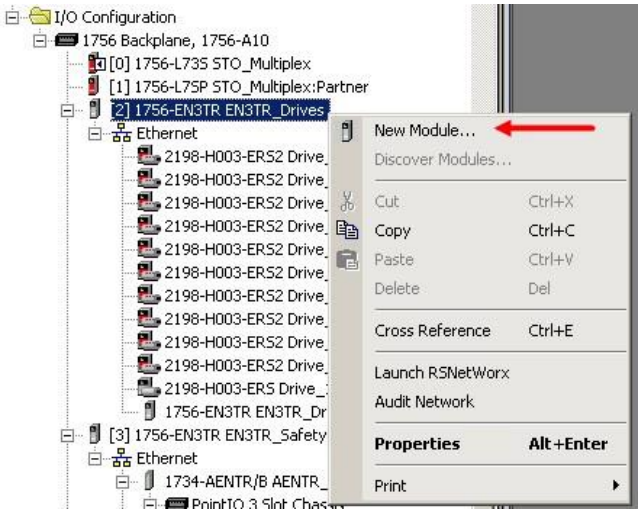


Configure a Network Safety Drive

Follow these steps to see how to configure Kinetix 5500 drives with networked STO.

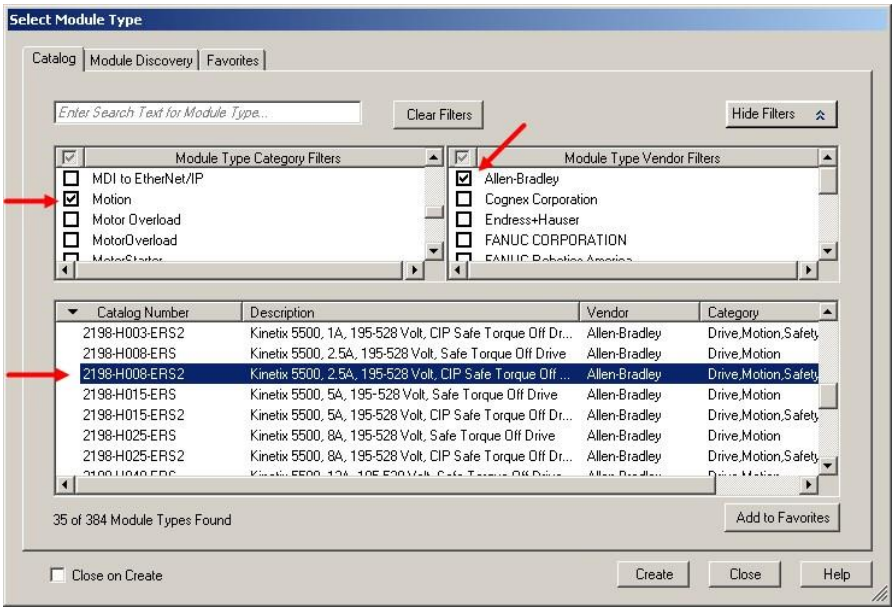
1. Open file **\Desktop\Lab Files\Network Safety Lab\Network_Safety_Begin.ACD**.

2. From the I/O tree, right-click on the **1756-EN3TR Module (EN3TR_Drives)** and choose **New Module…**.



The Select Module Type dialog appears.

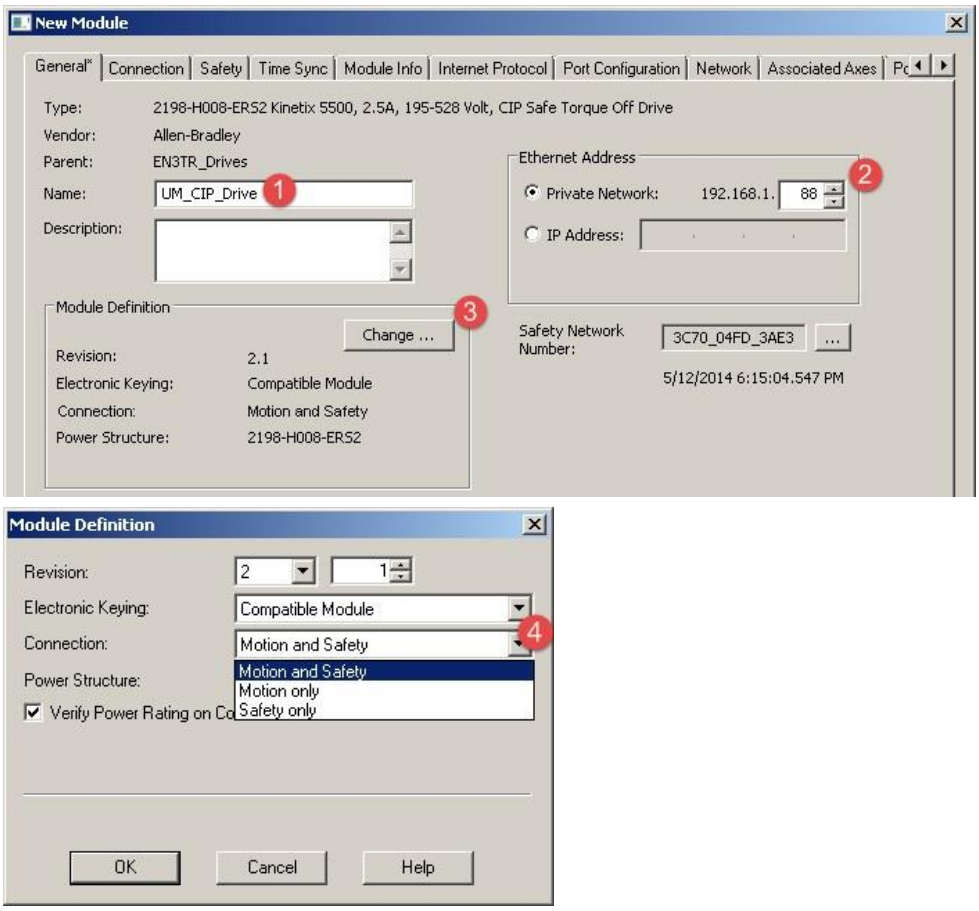
3. By using the filters, check **Motion and Allen-Bradley**, and select your **2198-H008-ERS2** servo drive.



4. Click the **Create** button.

   The New Module dialog box appears.

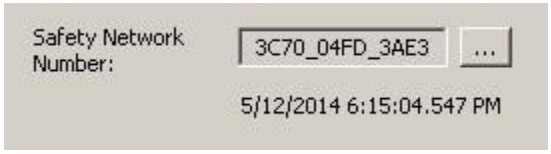
5. Configure the new drive.
   1. Type the drive Name: **UM_CIP_Drive**.
   2. Set Ethernet Address: **192.168.1.88**.
   3. Under Module Definition click Change. The Module Definition dialog box appears.
   4. From the Connection pull-down menu, choose the Connection mode; **Motion and Safety**



Note: When 'Safety' appears in the Connection mode, networked safety is implied.

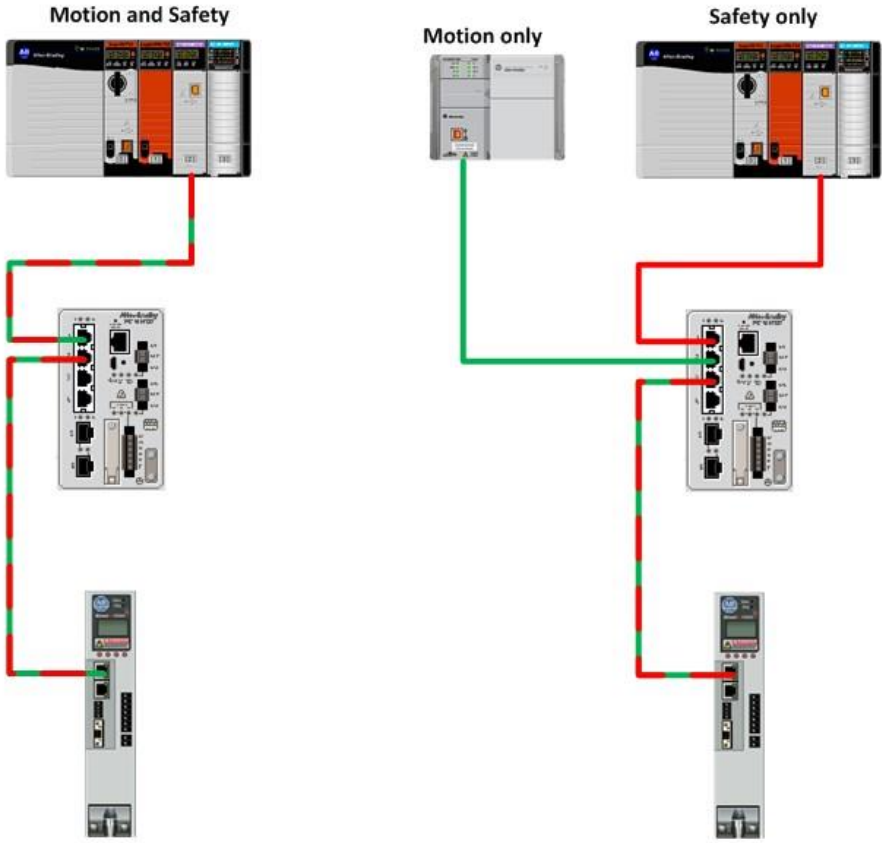
6. Click **OK** on the Module Definition dialog.

7. The Safety Network Number (SNN) field populates automatically when the Connection mode includes a networked Motion and Safety or Safety Only connection

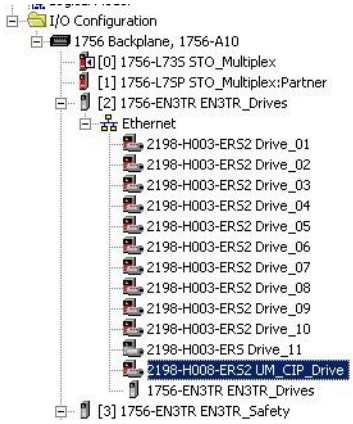


For a detailed explanation of the safety network number, refer to the GuardLogix Controller Systems Safety Reference Manual, publication 1756-RM099.

Connection Mode	Controller Needed	Description Drive Cat. No. 2198-Hxxx-ERS	Description Drive Cat. No. 2198-Hxxx-ERS2
Motion only	ControlLogix 1756L7x, GuardLogix 1756L7xS, or CompactLogix 5370	Only hardwired safe torque-off connections are possible.	Motion is managed by this controller. Safety is managed by another controller that has a Safety-only connection to the drive.
Motion and Safety	GuardLogix 1756-L7xS	N/A	Motion and Safety are managed by this controller.
Safety only	GuardLogix 1756-L7xS	N/A	Safety is managed by this controller. Motion is managed by another controller that has a Motion-only connection to the drive.



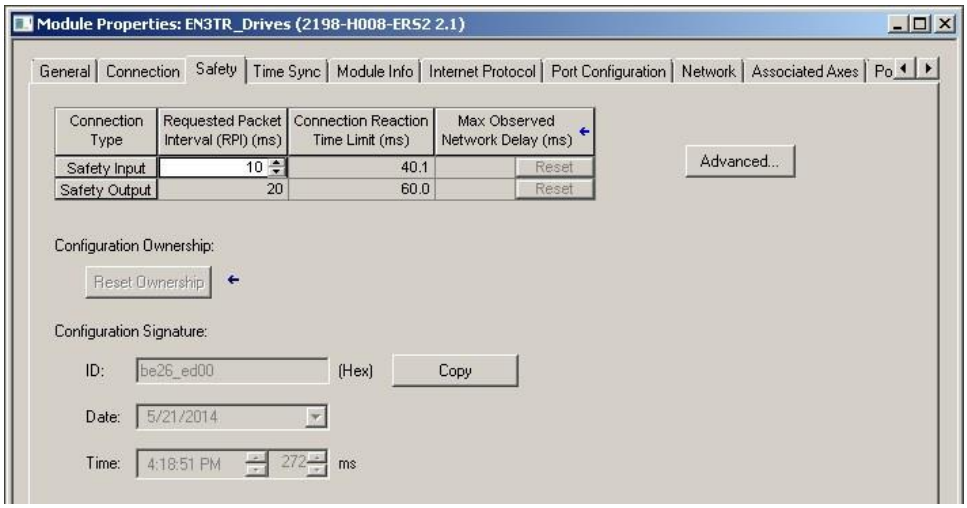
8. Click **OK** to close the New Module dialog box. Your 2198-H008-ERS2 servo drive appears in the Controller Organizer under the Ethernet controller in the I/O Configuration folder.



9. Right-click the drive you just created in the Controller Organizer and choose **Properties**.

   The Module Properties dialog box appears.

10. Click the **Safety** tab.



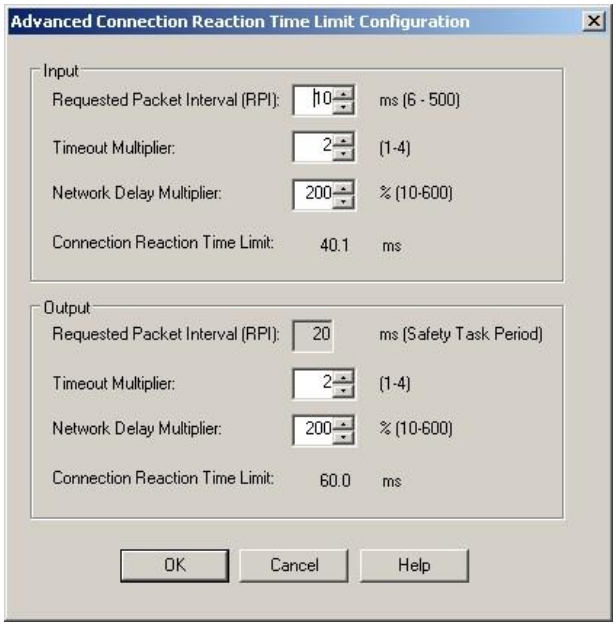
The connection between the owner and the 2198-Hxxx-ERS2 drive is based on the following:

- Servo drive catalog number must be 2198-Hxxx-ERS2 (networked)
- Servo drive safety network number
- GuardLogix slot number
- GuardLogix safety network number
- Path from the GuardLogix controller to the 2198-Hxxx-ERS2 drive
- Configuration signature

If any differences are detected, the connection between the GuardLogix controller and the 2198-Hxxx-ERS2 drive is lost, and the yellow yield icon appears in the controller project tree after you download the program.

11. Click **Advanced** button.

The Advanced Connection Reaction Time Limit Configuration dialog box appears.



Analyze each safety channel to determine the appropriate settings. The smallest Input RPI allowed is 6ms. Selecting small RPI values consumes network bandwidth and can cause nuisance trips because other devices cannot get access to the network.

12. Click **OK** to close the Advanced Connection Reaction Time Limit Configuration dialog box.

    For more information about the Advanced Connection Reaction Time Limit Configuration, refer to the GuardLogix 5570 Controllers User Manual, publication 1756-UM022.

13. Click **OK** to close the Module Properties dialog box.

Write Program Code

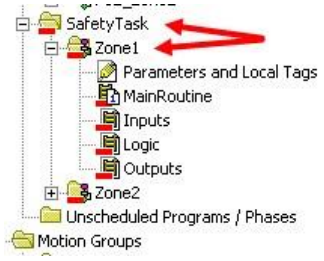
Let's examine the ladder logic associated with using networked Safe Torque Off drives, hardwired Safe Torque Off drives, and contactors. There are two zones in this example:

- Zone 1 has five network Safe Torque Off drives and one motor that is safeguarded with redundant contactors. This zone will utilize Stop Category 0, and coast to a stop upon a safety demand.
- Zone 2 has five network Safe Torque Off drives and one drive that is used in a hardwired configuration. This zone will utilize Stop Category 1, and ramp to stop upon a safety demand, removing power after a configurable time.

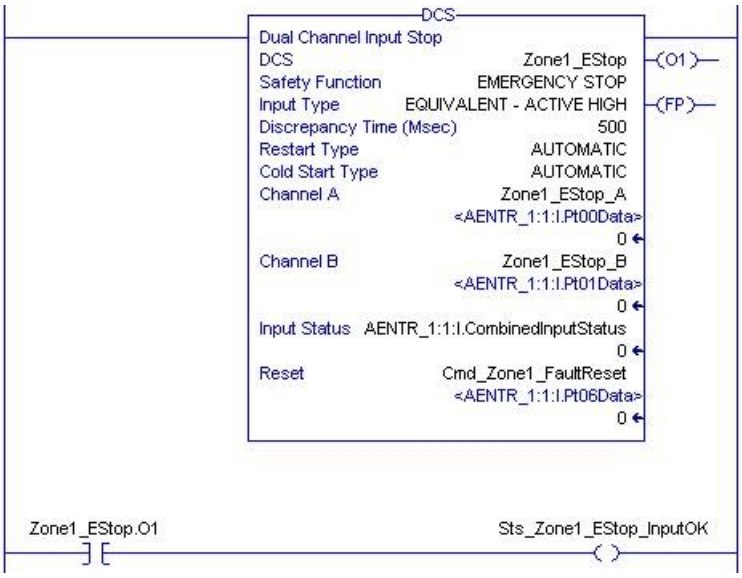
Each zone has the same inputs, including an Emergency Stop, a Light Curtain, and a SensaGuard door monitor. Each zone is represented as a program with routines for Input, Logic, and Output. The code in the Safety Task is based on ladder logic from the Safety Accelerator Toolkit and the standard task is based on the Drives and Motion Accelerator Toolkit.

Zone 1

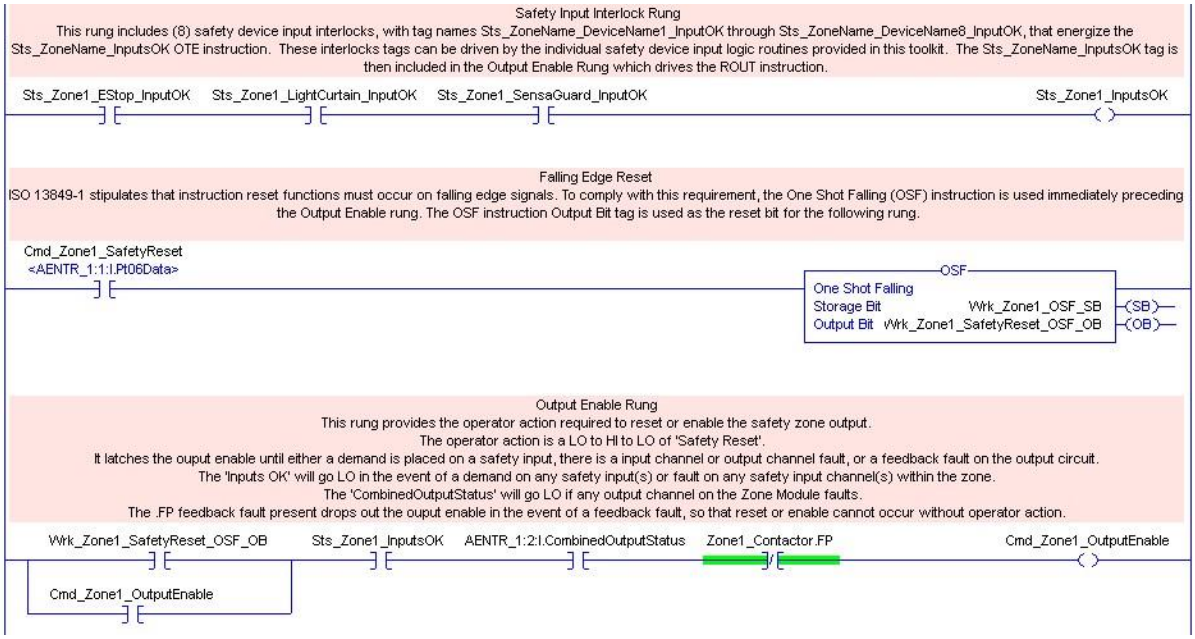
1. From the Safety Task in the Controller Organizer, expand the Zone1 program.



2. Review the Inputs routine. The three input devices are in this routine. The E-Stop code is shown here. There is extensive commentary in the rung descriptions that helps explain each portion of the code.

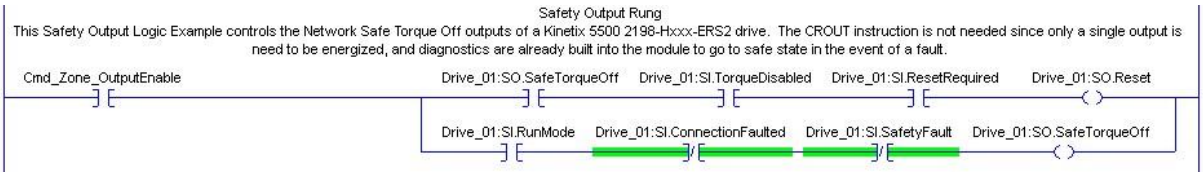


3. Review the Logic routine. These three rungs monitor the status of the inputs, restart functionality, and setting of the Output Enable bit. This logic is quite simple functionally (immediate removal of power), however there are more complex functions that can be developed as well.


Safety Input Interlock Rung
This rung includes (8) safety device input interlocks, with tag names Sts_ZoneName_DeviceName1_InputOK through Sts_ZoneName_DeviceName8_InputOK, that energize the Sts_ZoneName_InputsOK OTE instruction. These interlocks tags can be driven by the individual safety device input logic routines provided in this toolkit. The Sts_ZoneName_InputsOK tag is then included in the Output Enable Rung which drives the ROUT instruction.
Sts_Zone1_EStop_InputOK
Sts_Zone1_LightCurtain_InputOK
Sts_Zone1_SensaGuard_InputOK
Sts_Zone1_InputsOK
Falling Edge Reset
ISO 13849-1 stipulates that instruction reset functions must occur on falling edge signals. To comply with this requirement, the One Shot Falling (OSF) instruction is used immediately preceding the Output Enable rung. The OSF instruction Output Bit tag is used as the reset bit for the following rung.
Cmd_Zone1_SafetyReset
<AENTR_1:1:I.Pt06Data>
OSF
One Shot Falling
Storage Bit
Wrk_Zone1_OSF_SB
Output Bit
Wrk_Zone1_SafetyReset_OSF_OB
SB
OB
Output Enable Rung
This rung provides the operator action required to reset or enable the safety zone output.
The operator action is a LO to HI to LO of 'Safety Reset'.
It latches the ouput enable until either a demand is placed on a safety input, there is a input channel or output channel fault, or a feedback fault on the output circuit.
The 'Inputs OK' will go LO in the event of a demand on any safety input(s) or fault on any safety input channel(s) within the zone.
The 'CombinedOutputStatus' will go LO if any output channel on the Zone Module faults.
The .FP feedback fault present drops out the ouput enable in the event of a feedback fault, so that reset or enable cannot occur without operator action.
Wrk_Zone1_SafetyReset_OSF_OB
Sts_Zone1_InputsOK
AENTR_1:2:I.CombinedOutputStatus
Zone1_Contactor.FP
Cmd_Zone1_OutputEnable
Cmd_Zone1_OutputEnable

4. In the Outputs routine from Zone 1, there are two different examples. The first five devices are all Network Safety drives, while the last example is a contactor.

   The drives have much simpler code because they handle all of their own diagnostics and can easily report back that information to the controller, as shown:


Safety Output Rung
This Safety Output Logic Example controls the Network Safe Torque Off outputs of a Kinetix 5500 2198-Hxxx-ERS2 drive. The CROUT instruction is not needed since only a single output is need to be energized, and diagnostics are already built into the module to go to safe state in the event of a fault.
Cmd_Zone_OutputEnable
Drive_01:SO.SafeTorqueOff
Drive_01:SI.TorqueDisabled
Drive_01:SI.ResetRequired
Drive_01:SO.Reset
Drive_01:SI.RunMode
Drive_01:SI.ConnectionFaulted
Drive_01:SI.SafetyFault
Drive_01:SO.SafeTorqueOff

Note: This could even be combined into a simple Add-On Instruction for even more simplicity!

5. The last two rungs of the Outputs routine from Zone 1 demonstrate the additional work that needs to be included for contactors. The controller must manage all of the diagnostics for the contactors, so the CROUT instruction is used to coordinate the timing of the actuation command, feedback, and module statuses.

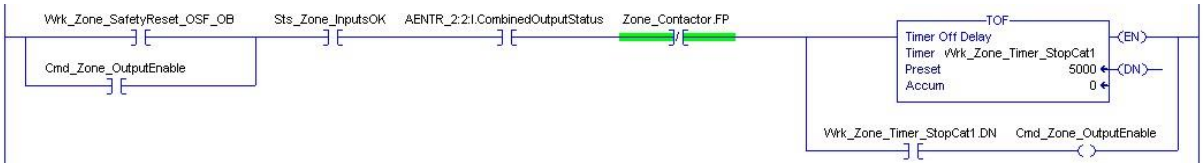
Zone 2

6. From the Safety Task in the Controller Organizer, expand the Zone2 program.

7. Since the Input routine is similar to Zone1, skip ahead and open the **Logic** routine.

8. There is an important difference in this routine on the rung (rung 2) that energizes the Output Enable bit.

   The addition of the TOF instruction gives the standard task time to execute stopping instructions to put the axes into a disabled state at a known position before the torque is removed. This is essential for vertical loads and many other coordinated applications.



9. Open up routine **MainTask -> P02_Zone2 -> R03_Control** and examine rung 3.

   Since these drives are only rated for Stop Category 0, the programmer should plan to execute code in the Standard Task to bring the drives to a stop and disable them before the torque is removed. This ensures that any mechanical brakes can be set before holding torque disappears.

   The addition of the "\Zone2.Sts_Zone_InputsOK" tag provides a "Stop" command to the application. This will stop the running sequence and reset sequence, and initiate the stopping sequence. By doing this, you can program the machine to come to the controlled stop of your desire.

10. Most of the Outputs routine remains unchanged. There is a difference in the last two rungs from Zone 1. The Feedback parameters for the CROUT instruction are tied to tags mapped from the Standard Task to the Safety Task. Open the routine and view this difference in the last two rungs.

11. Tag mapping is accomplished from the dialog box that appears after following the menu path; **Logic -> Map Safety Tags**.

12. Follow the path and view the dialog box.

Feedback for purely diagnostic purposes is a common function that uses mapping from the Standard Task to the Safety Task. Reset functionality does not necessarily need to be "safety rated", since many other safeguards are in place to prevent restart when dangerous situations could occur, and represents another example of when to use Tag Mapping. Tag mapping should not be abused, since putting logic in the Safety Task does not necessarily make it "safe", but it can be a very helpful tool for appropriate uses.

13. Close the dialog box when finished.

14. To see how the mapped tag is energized, open up the **DriveManagerTask -> P11_Axis_11 -> R02_Monitor** routine and look at **rung 24**. The Servo_Axis.GuardGateDriveOutputStatus tag is used to reflect back to the Kinetix_STO_Feedback_Map tag, the status of the gate drivers in the servo drive.

This concludes this lab.

Appendix F: Multiplexing Introduction

The goal of this lab is to provide an overview of Drive Mulitplexing, a new feature for an upcoming release of Studio 5000.

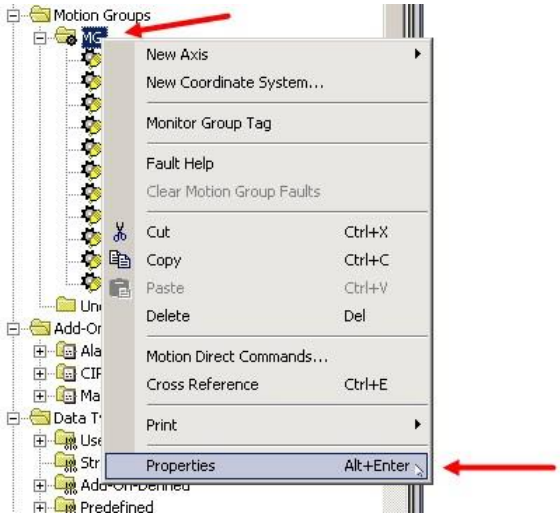
Multiplexing allows for the controller to execute up to three effective Coarse Update Periods for the Motion Group. There is still only one Motion Group with multiplexing, but drives can execute at different rates, relative to their performance requirements. Previously, a single high performance axis could require all of the other axis on the controller to operate at the same rate. This leads to extended scan times for other logic, since the Motion Group has the highest priority in the controller. An arrangement like this could also lead to requiring an additional controller to handle some of the high performance motion axes, since the low performance axes have to be updated faster than the application requires.

Using Multiplexing to Optimize Performance

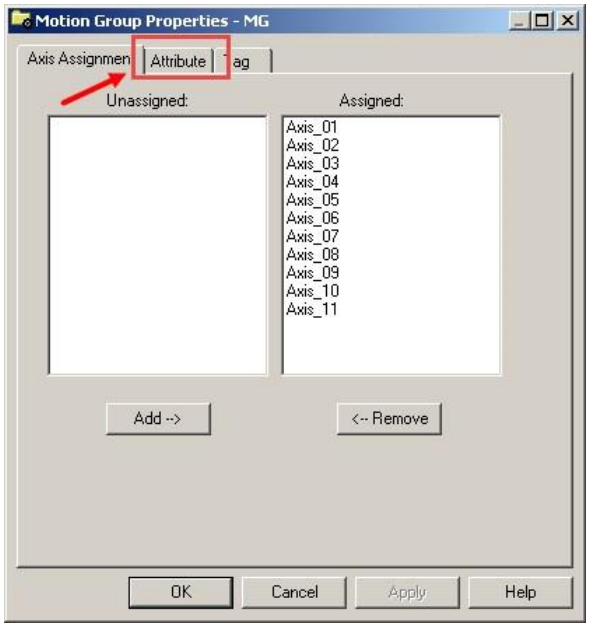
In this section, you will use a project that has many axes, and is near its utilization limits. You will see how to optimize performance for different axes and scan times, to reduce the burden on the controller and get more processing power for the same cost.

1. Open file **\Desktop\Lab Files\Multiplexing Lab\Multiplexing_Begin.ACD**.

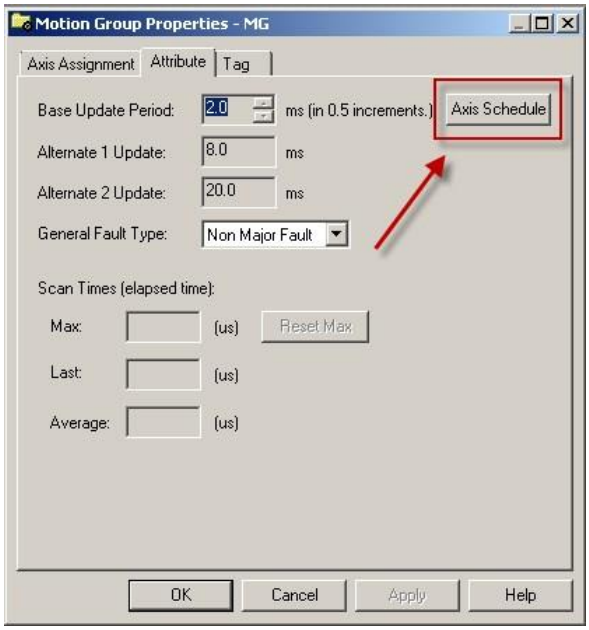
2. Locate the Motion Group **MG** in the Controller Organizer, and select **Properties**.



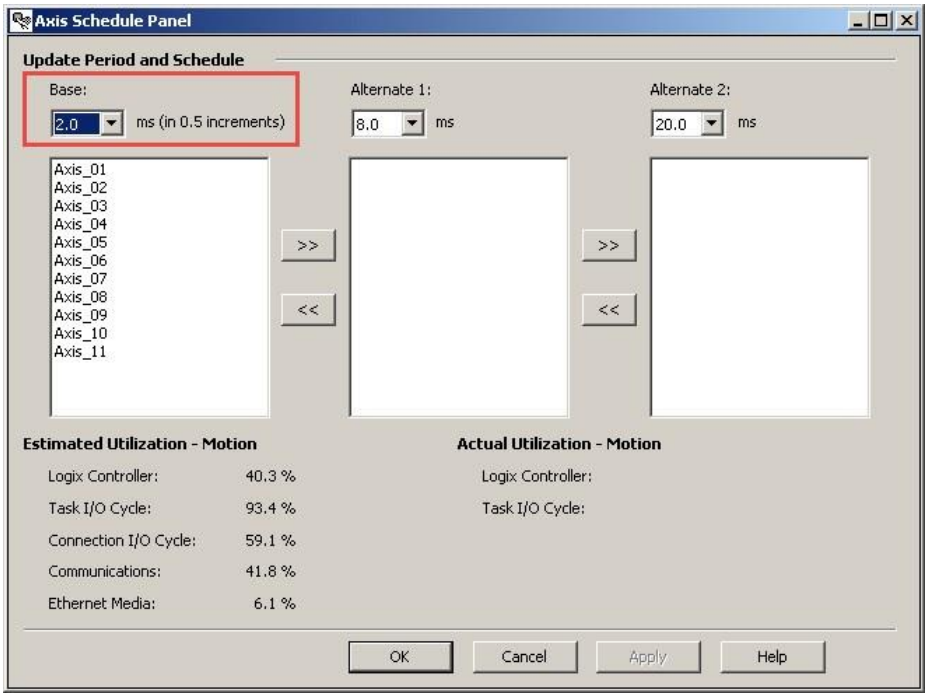
3. The dialog box will appear, and you should select the Attribute tab.



4. Select the **Ellipsis** button shown below to access the **Axis Schedule Panel**.

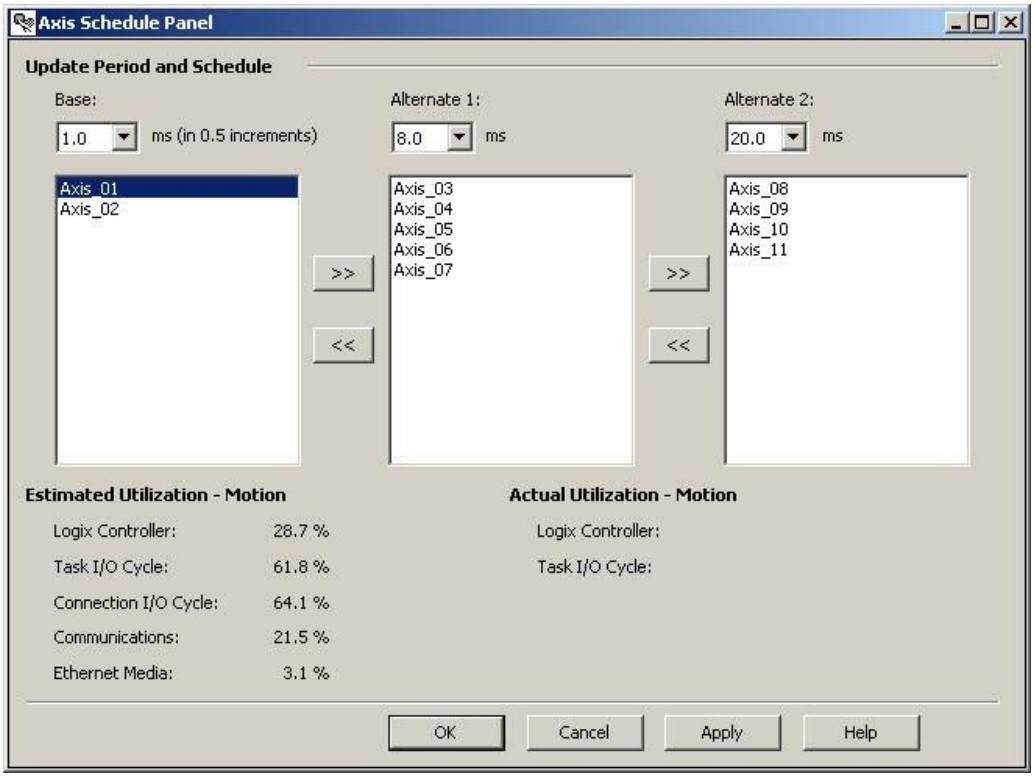


5. Within the Axis Schedule Panel, you can adjust which axes fall into each of the Update Schedules, as well as view the estimated utilization of some key metrics. In the screen shown, the Base Update Rate is 2.0ms.



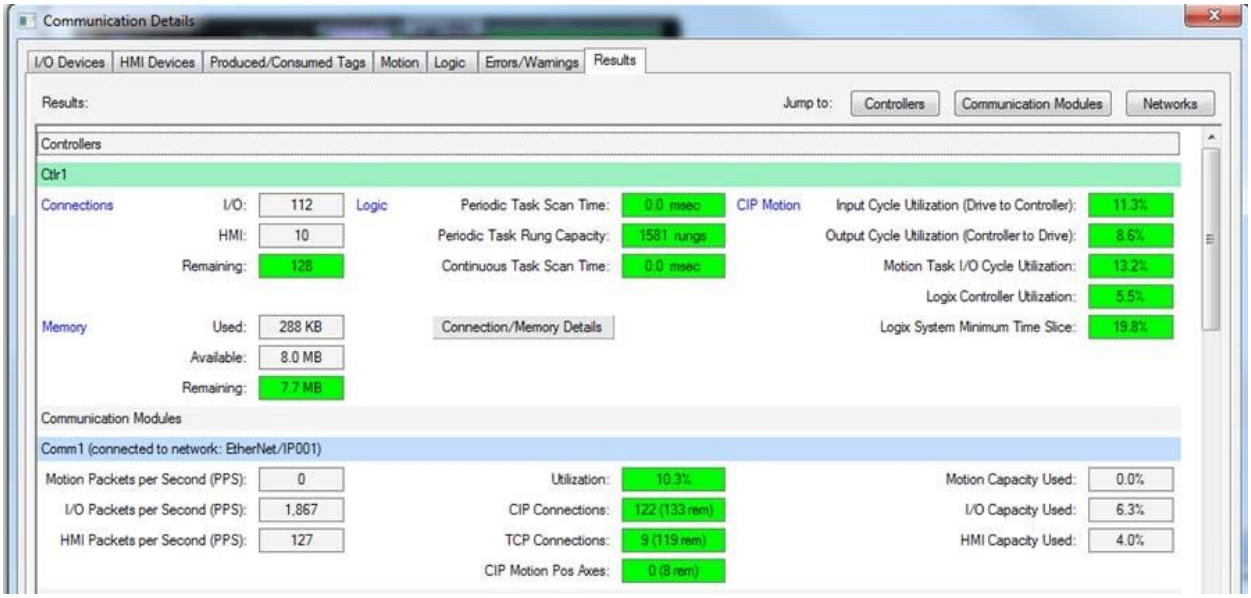
6. Imagine that you just discovered that Axis_01 and Axis_02 require a faster update rate to keep up with a very demanding application. Try setting the **Base Update Rate** to **1.0ms** and see what happens to the system estimated utilization.

7. You may have noticed warning symbols, indicating that task overlaps may occur, or that too much Logix processing is being spent on Motion Planning. This would impact other tasks within the controller. For this application, let's look at a few other axes though. Axis_03 through Axis_07 are servo applications that can be run at a slower rate than our "high performance" application, and Axis_08 through Axis_11 are simply running conveyors and don't need to update fast at all. Try to optimize the Update Rates, and group schedules to maintain tight control over Axis_01 and Axis_02, without jeopardizing any of the key utilization metrics. An example configuration is shown below.



IAB Info

This section is not a comprehensive view of system sizing however. It is recommended to attend any labs that are offered for Integrated Architecture Builder for a more information on network and controller sizing. That program includes a sizing tool using the same algorithms, and allows for complex system design including HMI tags, I/O devices, and other communications. An example of the results screen for the sizing tool in Integrated Architecture Builder is shown below.



This concludes this lab.

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe/Middle East/Africa: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication CE-DM001-EN-P– Dec. 2014
Supersedes Publication CE-DM001-EN-P Oct. 2014

Copyright© 2014 Rockwell Automation, Inc. All rights reserved.